



Gestión de Recursos Humanos en Proyectos Informáticos

Universidad de Castilla-La Mancha
Escuela Superior de Informática
Planificación y Gestión de Sistemas de Información
Francisco Ruiz González
Miguel Angel Molina Tejedor
Ciudad Real, 15 de Mayo del 2000

Índice

1.- INTRODUCCIÓN	4
2.- SELECCIÓN, FORMACIÓN, DIRECCIÓN.....	16
2.1.- SELECCIÓN	16
2.1.1.- <i>El proceso de convocatoria.....</i>	16
2.1.2.- <i>Etapas de preselección.....</i>	18
2.1.3.- <i>El proceso de selección</i>	18
2.1.4.- <i>Toma de decisión e incorporación</i>	19
2.2.- FORMACIÓN	20
2.2.1.- <i>Estimación de necesidades</i>	20
2.3.- DIRECCIÓN.....	22
2.3.1.- <i>¿Quién manda aquí?.....</i>	22
2.3.2.- <i>Responsabilidades del director de proyecto</i>	23
2.3.3.- <i>Los estilos de dirección.....</i>	24
2.3.4.- <i>Directivos y responsables técnicos</i>	26
3.- LA INFLUENCIA DE LA PLANIFICACIÓN.....	27
3.1.- PLANIFICACIÓN EXCESIVAMENTE OPTIMISTA	27
3.1.1.- <i>Causas fundamentales de las planificaciones demasiado optimistas</i>	27
3.1.2.- <i>Efectos de la planificación excesivamente optimista</i>	27
3.2.- PRESIÓN EXCESIVA EN LA PLANIFICACIÓN	29
3.2.1.- <i>Víctimas de la presión excesiva en la planificación</i>	30
3.2.2.- <i>Puntos cruciales.....</i>	31
3.3.- DISMINUCIÓN DE LA PRESIÓN DE LA PLANIFICACIÓN	33
4.- MOTIVACIÓN	35
4.1.- MOTIVACIONES TÍPICAS DEL DESARROLLADOR	35
4.2.- USO DE LOS CINCO FACTORES DE MOTIVACIÓN	37
4.2.1.- <i>Realización.....</i>	37
4.2.2.- <i>Posibilidad de superación</i>	37
4.2.3.- <i>El trabajo en sí</i>	38
4.2.4.- <i>Vida personal</i>	39
4.2.5.- <i>Oportunidad de supervisión técnica.....</i>	39
4.3.- USO DE OTROS FACTORES DE MOTIVACIÓN	40
4.3.1.- <i>Premios e incentivos</i>	40
4.3.2.- <i>Proyectos piloto.....</i>	41
4.3.3.- <i>Revisiones del rendimiento.....</i>	41
4.4.- DESTRUCTORES DE LA MORAL.....	42
4.4.1.- <i>Factores de higiene.....</i>	42
4.4.2.- <i>Otros destructores de la moral.....</i>	42
5.- EL EQUIPO DE TRABAJO	45
5.1.- USOS DE LOS EQUIPOS DE TRABAJO EN EL SOFTWARE	45
5.2.- IMPORTANCIA DEL EQUIPO DE TRABAJO.....	45
5.3.- CREACIÓN DE UN EQUIPO DE ALTO RENDIMIENTO	46
5.3.1.- <i>Características de un equipo de alto rendimiento.....</i>	47
5.3.2.- <i>Dirección de un equipo de alto rendimiento.....</i>	50
5.4.- CAUSAS DE FALLO DE LOS EQUIPOS.....	51
5.5.- CONFIGURACIÓN DE EQUIPOS A LARGO PLAZO.	53
5.6.- RESUMEN DE DIRECTRICES PARA GRUPOS DE TRABAJO	54
6.- ESTRUCTURA DEL EQUIPO	55
6.1.- CONSIDERACIONES SOBRE LA ESTRUCTURA DEL EQUIPO	55
6.1.1.- <i>Tipos de equipos.....</i>	55
6.1.2.- <i>Características adicionales para diseño de equipos.....</i>	56
6.1.3.- <i>¿Cuál es el mejor tipo de equipo para un desarrollo rápido?</i>	57
6.2.- MODELOS DE EQUIPO	57

6.2.1.- Equipo de negocios.....	57
6.2.2.- Equipo con programador jefe o quirúrgico.....	58
6.2.3.- Equipo en la sombra	59
6.2.4.- Equipo de prestaciones.....	59
6.2.5.- Equipo de emergencias o de búsqueda y rescate.	60
6.2.6.- Equipo de especialistas (G.E.O.).....	60
6.2.7.- Equipo deportivo	60
6.2.8.- Equipo de teatro.....	61
6.3.- GRANDES EQUIPOS.....	61
6.4- MALA COMUNICACIÓN	62
6.4.1.- La comunicación como fin y no como medio	62
6.4.2.- Arterioesclerosis de la información	63
6.4.3.- Mensajes distorsionados.....	63
7.- CONCLUSIONES	64
FUENTES Y BIBLIOGRAFÍA	69
ANEXO I. MÉTODOS RECOMENDABLES	73
I.1.- INTRODUCCIÓN A LOS MÉTODOS RECOMENDABLES	73
I.2.- ENTORNOS PRODUCTIVOS	75
I.2.1.- Uso de entornos productivos	76
I.2.2.- Gestión de los riesgos de los entornos productivos	76
I.2.3.- Efectos secundarios de los entornos productivos.....	77
I.2.4.- Interacción de los entornos productivos con otros métodos.....	77
I.2.5.- Puntos cruciales de los entornos productivos.....	78
I.2.6.- Claves para el éxito en el uso de entornos productivos	78
I.3.- COMPROMISO (SIGNING UP)	79
I.3.1.- Uso del compromiso.....	79
I.3.2.- Gestión de los riesgos del compromiso	80
I.3.3.- Efectos secundarios del compromiso	81
I.3.4.- Interacciones del compromiso con otros métodos.....	81
I.3.5.- Puntos cruciales del compromiso	81
I.3.6.- Claves para el éxito en el uso del compromiso	81
I.4.- GESTIÓN THEORY-W	82
I.4.1.- Uso de la gestión Theory-W.....	83
I.4.2.- Gestión de los riesgos de la gestión Theory-W	85
I.4.3.- Efectos secundarios de la gestión Theory-W	86
I.4.4.- Interacciones de la gestión Theory-W con otros métodos.....	86
I.4.5.- Puntos cruciales de la gestión Theory-W	86
I.4.6.- Claves para el éxito en el uso de la gestión Theory-W	86
I.5.- HORAS EXTRAS VOLUNTARIAS	87
I.5.1.- Uso de las horas extras voluntarias	87
I.5.2.- Gestión de los riesgos de las horas extras voluntarias.....	89
I.5.3.- Efectos secundarios de las horas extras voluntarias.....	90
I.5.4.- Interacciones de las horas extras voluntarias con otros métodos.....	90
I.5.5.- Puntos cruciales de las horas extras voluntarias.....	90
I.5.6.- Claves para el éxito en el uso de las horas extras voluntarias	90
ANEXO II. CONTRAINDICACIONES DE ESTRATEGIAS ALTERNATIVAS	91
ANEXO III. TEST MYERS-BRIGGS	93
III.1.- ANTECEDENTES DEL MÉTODO	93
III.2.- DIMENSIONES.....	94
III.2.1.- La dimensión extroversión - introversión	94
III.2.2.- La dimensión sensibilidad - intuición.....	95
III.2.3.- La dimensión racional - afectividad.....	95
III.2.4.- La dimensión juicio - percepción.....	95
III.3.- APLICAR A LOS PROYECTOS LA TEORÍA DE LOS TIPOS PSICOLÓGICOS	96
ANEXO IV. PREVENCIÓN DE RIESGOS LABORALES.	97

1.- Introducción

¿Gestión de Recursos Humanos en Proyectos Informáticos?, Bonita concatenación de palabras, ¿Qué es todo esto? Según el Diccionario de la Real Academia de la Lengua [RAE, 1995]:

gestión.

Del lat. gestio, -onis.

1. f. Acción y efecto de gestionar.

[. . .]

gestionar.

De gestión.

1. tr. Hacer diligencias conducentes al logro de un negocio o de un deseo cualquiera.

O sea, hacer unos planteamientos y llevar a cabo las conclusiones que se deduzcan de los mismos orientadas a:

recurso.

Del lat. recursus.

[. . .]

7. [m.] Conjunto de elementos disponibles para resolver una necesidad o llevar a cabo una empresa. RECURSOS naturales, hidráulicos, forestales, económicos, humanos, etc.

[. . .]

humano, na.

Del lat. humanus.

1. adj. Perteneciente al hombre o propio de él.

[. . .]

9. m. pl. Conjunto de todos los hombres.

Con lo cual sabemos que contamos con una serie de personas para poder realizar nuestro

proyecto, ta.

Del lat. proiectus.

[. . .]

4. [m.] Designio o pensamiento de ejecutar algo.

5. [m.] Conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo ha de ser y lo que ha de costar una obra de arquitectura o de ingeniería.

6. [m.] Primer esquema o plan de cualquier trabajo que se hace a veces como prueba antes de darle la forma definitiva.

[. . .]

informático, ca.

1. adj. Perteneciente o relativo a la informática.

2. [adj.] Que trabaja o investiga en informática. Apl. a pers., ú. t. c. s.

Por lo tanto una finalidad, un conjunto de metodologías y un medio (el informático).

Con lo que ya tenemos que:

Partiendo de un planteamiento debemos aplicar mediante unas metodologías una serie de conclusiones a un equipo humano para conseguir (haciendo uso de la informática) una finalidad.

Nos intentaremos centrar en el planteamiento, la obtención de las conclusiones y el conocimiento de algunas metodologías así como de su aplicación. Teniendo siempre en cuenta que muchos de los temas críticos no son prácticos, sino más bien estratégicos y a veces filosóficos. En ocasiones no hay nada más práctico que una buena teoría. Además diferentes proyectos tienen diferentes necesidades y no existen métodos mágicos.

Mucho se ha hablado, escrito y discutido sobre gestión de proyectos en general y de proyectos informáticos en particular, la tendencia hace unos años se centraba en el análisis, diseño, estilo de codificación, estimación de costes, etc. se generaban montones y montones de diagramas, gráficos, montañas de papel y *megas* de información. Se planificaba, se daban unos plazos y todo entraba en una caja negra de la que al final salía el producto. Un producto que en muchas ocasiones no satisfacía al cliente, en el que los plazos se cumplían tarde, mal y nunca, rara vez con una calidad aceptable y donde los costes se disparaban en todos los casos.

Esto no es nada nuevo ni por desgracia perteneciente al pasado, esta muy bien planificar, estimar, comunicar con metodologías más o menos estándar pero hasta hace relativamente poco los equipos de análisis y desarrollo (más estos últimos) no eran sino gente con un ordenador dentro de una campana de cristal a los que dar unas pautas para algún día obtener algo parecido a lo que se pedía.

Afortunadamente y como no podía ser de otra manera las empresas se han dado cuenta de que descuidar el factor humano en un proyecto es un grave error. La gestión de los recursos humanos en un proyecto es clave y vital, tan importante como un buen análisis.

Ahora se selecciona, forma, incentiva, etc. al trabajador, se agilizan los canales de comunicación y se flexibiliza la metodología. Tratar el tema de los recursos humanos no es tarea fácil máxime cuando se centra en un proyecto informático. Es muy sencillo caer en planteamientos sociológicos y filosóficos¹.

En los proyectos y en general la solución de clientes y directivos ante problemas de tiempo en desarrollo es incrementar la presión de planificación y las horas extras de los programadores. El exceso de presión de planificación en alrededor del 75 por 100 de todos los grandes proyectos, y se acerca al 100 por 100 en los proyectos muy grandes [Jones, 1994]. Prácticamente el 60 por 100 de los desarrolladores comentan que su nivel de estrés aumenta progresivamente [Glass, 1994c]. En Estados Unidos, el desarrollador medio trabaja de 48 a 50 horas a la semana [Zawacki, 1993], y en un momento en el que la industria necesita desesperadamente reclutar nuevos programadores para suavizar la presión de planificación, a los desarrolladores les deja de parecer divertido su campo.

Como veremos más adelante de nada vale incrementar la presión, debe existir una planificación.

¹ N.D.A. planteamientos en los que me permitiré caer en la introducción e intentaré evitar en adelante.

Si cogiésemos 100 músicos de prestigio a nivel mundial y los pusiésemos en una orquesta sin director, no sonarían como una orquesta de calidad. El tempo de la sección de cuerda no concordaría con el de la sección de madera o con la sección de metal. Indicar a los músicos que lo hagan “lo mejor que puedan” no les ayuda a saber si deben ir más rápido o más lento, Semejante evento musical sería un desperdicio de talento [Mc Connell, 1996].

Por otra parte debemos situar los recursos humanos y la gestión de los mismos en un contexto. La mayoría de los proyectos informáticos tienen como finalidad crear y gestionar sistemas de información (SI) automatizados, y como empresa de desarrollo el origen también es un sistema de información.

Cuanto más grande y complejo es un sistema, más importante es la comunicación de información entre sus componentes e implicados.

Los elementos de un SI son [De Pablo, 1989] y [Alter, 1992]:

- Los procedimientos y prácticas habituales de trabajo.
- La información (elemento principal del sistema).
- **Las personas o usuarios.**
- El equipo de soporte (constituye la parte más visible).

Las maneras y prácticas de trabajo marcan:

- Que información se necesita.
- **Como se ven implicadas las personas.**
- Que equipo se precisa para un trabajo eficaz.

Las prácticas de trabajo obligan a la gente a adoptar una cierta manera de actuar, pero deben, a su vez, adaptarse a las características de las personas implicadas para ser eficaces.

La estructura de un SI esta compuesta por varios subsistemas:

- **Subsistema de recursos humanos.**
- Subsistema de gestión contable.
- Subsistema de gestión comercial.
- Subsistema de control de existencias.

No hay que profundizar más en el tema pues solo se quiere aclarar parte del contexto, no obstante si nos atañe y es interesante comentar algo más sobre el subsistema de recursos humanos.

El recurso o activo más importante en una empresa es el constituido por el conjunto de personas que la componen. Las actividades de gestión relacionadas con el personal de la empresa se basan en dos aspectos principales:

1. **La gestión de la información relacionada con la plantilla.** Que incluye información personal compuesta de:

- Filiación completa.
- Datos médicos.
- Historial laboral.
- Datos relacionados con el salario y los incentivos.
- Datos sobre la carrera profesional y el historial formativo.

El nivel de mayor complejidad lo representa la posibilidad de realizar una planificación y optimización de la plantilla en función de los objetivos de la empresa, lo que implica el manejo y el análisis de la estructura organizativa. Así mismo, el sistema debería dar soporte al proceso de reclutamiento de nuevos empleados.

2. La ejecución de la nomina.

El subsistema de recursos humanos es el que más cambios sufre como reacción a los cambios del entorno.

La gestión de los recursos humanos ejerce sus actividades en todos los niveles de la jerarquía de la empresa. A nivel operativo se responsabiliza de:

- Mantenimiento de datos de los empleados.
- Inventario de cualificaciones de los empleados.
- Inventario de puestos de trabajo existentes en la empresa y de las condiciones más adecuadas para desempeñarlos.
- Evaluación de los empleados.
- Gestión de las solicitudes de empleo.

En el nivel intermedio de decisiones el subsistema de gestión de recursos humanos debería dar soporte a las siguientes actividades:

- Análisis y diseño del perfil de persona ideal para cada puesto de trabajo, para saber que tipo de persona hay que contratar o que formación debe recibir alguien antes de ascender a un cargo.
- Analizar las necesidades de contratación de personal para cumplir los objetivos de la empresa a medio plazo, generando planes de previsión de reclutamiento de plantilla.
- Generar planes para crear incentivos y aplicar beneficios sociales a los empleados.
- Análisis de las necesidades de formación y creación de planes para la mejora del nivel técnico profesional de la plantilla.

Por último, a nivel estratégico, la gestión de recursos humanos, debe preocuparse de crear planes que indiquen la cantidad de personal, sus características, las fechas en las que contratar, los cursos de formación necesarios, etc. para poder contar con la infraestructura de personal apropiada para los objetivos de la empresa.

Las organizaciones realizan trabajo. El trabajo generalmente aglutina operaciones o proyectos, ambos conceptos se mezclan. Las operaciones y los proyectos comparten algunas características, por ejemplo:

- **Están hechos por personas.**
- Delimitados por recursos finitos.
- Se planifican, ejecutan y controlan.

Una de las claves vitales de cualquier proyecto son los objetivos. Para establecer objetivos realistas hay que hacer que las personas que quieren que el trabajo se haga y las personas que harán el trabajo desarrollen los objetivos conjuntamente. Este procedimiento de establecimiento de metas asegura cierto grado de compromiso de todas las partes.

La necesidad de compromiso se pone de evidencia de manera importante al finalizar el ejercicio de fijación de objetivos, porque en ese punto el equipo de gestión y el equipo de desarrollo firman un “contrato” de MBO, en el que la dirección expresa su compromiso de apoyar el trabajo, y los desarrolladores demuestran su voluntad de realizar ese trabajo.

Un problema muy común es que las personas suelen quedarse empantanadas en la negociación de los objetivos. Suele suceder que los actores dediquen más tiempo a definir los objetivos que a realizar el trabajo.

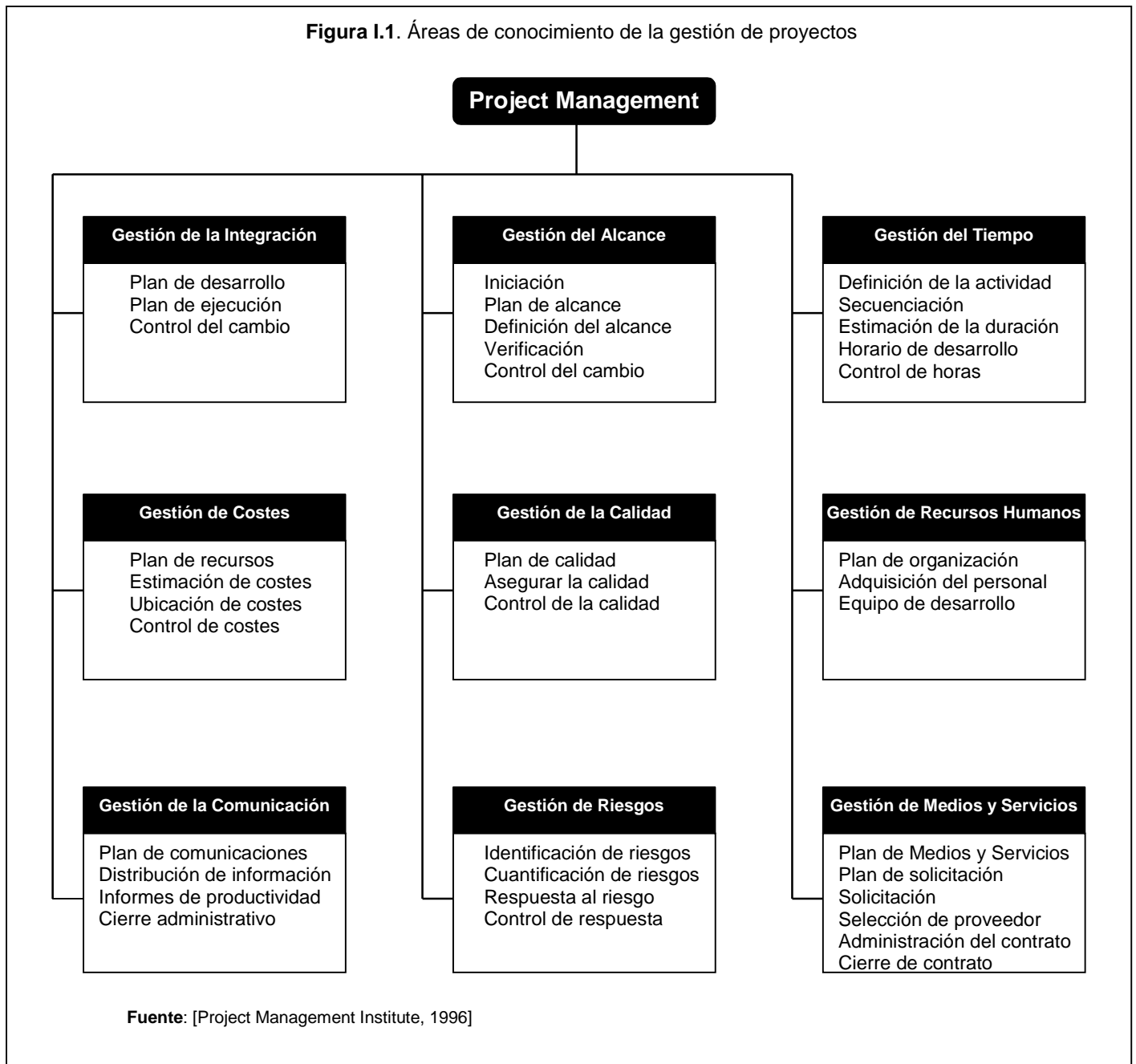
En cuanto a la gestión de proyectos; *es la aplicación del conocimiento, experiencia, herramientas y técnicas a las actividades de los proyectos para unificar o separar las necesidades de los implicados y la expectativa de un proyecto* [Project Management Institute, 1996]. Juntar necesidades y expectativas invariablemente reclama un balance competitivo entre:

- Alcance, tiempo, coste y calidad.
- **Diferentes necesidades y expectativas de los implicados.**
- Requisitos identificados (necesidades) y requisitos no identificados (expectaciones).

Las áreas de la gestión de proyectos, describen el conocimiento y la práctica de la gestión de proyectos en función de los procesos que la componen. Estos procesos están organizados según The Project Management Institute [1996] en las nueve áreas descritas a continuación e ilustradas en la figura I.1.

- **Gestión de la Integración**, describe los procesos necesarios para asegurar que los distintos elementos de un proyecto están correctamente coordinados.
- **Gestión del alcance**, procesos para asegurar que el proyecto incluye todo el trabajo requerido, y solo el requerido, para completar el proyecto con éxito.
- **Gestión del tiempo**, asegura la finalización dentro de plazo.
- **Gestión de costes**, permitirá cerrar el proyecto con el presupuesto aprobado.
- **Gestión de la calidad**, satisfacer las necesidades para las que el proyecto se penso.
- **Gestión de los recursos humanos**, para hacer más efectivo el uso de la gente involucrada en el proyecto.
- **Gestión de la comunicación**, describe los procesos necesarios para asegurar tiempos, colección, diseminación, almacenamiento, etc. de la información del proyecto.
- **Gestión de riesgos**, identificar, analizar y responder a los riesgos del proyecto.
- **Gestión de medios y servicios**, procesos necesarios para conseguir y gestionar recursos, medios y servicios externos.

Figura I.1. Áreas de conocimiento de la gestión de proyectos



La planificación es otro punto importante. El plan es un mapa de ruta que nos indica como ir de un punto a otro. Una vez que se ha decidido llevar a cabo un proyecto, empieza una detallada planificación formal. Se identifican los hitos del proyecto, y se fijan las tareas y su interdependencia. A medida que el proyecto avanza, el plan puede sufrir continuas modificaciones, que reflejaran las circunstancias imprevistas que se presenten y las respuestas que se les dé. Rara vez los planes de proyecto son formulaciones estáticas, son, por el contrario instrumentos dinámicos que permiten al equipo del proyecto manejar el cambio ordenadamente. De hecho, es preciso señalar que todos los planes son, en alguna medida, suposiciones.

El ciclo de vida, condicionará la gestión de un proyecto y de los recursos humanos asignados al mismo. El ciclo de vida de un proyecto define:

- Que trabajo técnico debería hacerse en cada fase.
- **Quien y que debería implicarse en cada fase.**

La influencia de los implicados en el proyecto en las características finales del producto y del coste final del proyecto, es alta al principio y disminuye a medida que avanza el proyecto. Este fenómeno hace que los costes de los cambios y la corrección de errores se incrementen durante la realización del proyecto y las distintas fases del ciclo vital.

En un proyecto puede haber tanto individuos como organizaciones o empresas, cuyos intereses pueden afectar positiva o negativamente a la ejecución del proyecto o a un resultado satisfactorio o no del mismo. El equipo de dirección y gestión de un proyecto debe identificar a los implicados, determinar cuales son sus necesidades y expectativas y gestionar las mismas (tarea de especial dificultad) para asegurar el éxito del proyecto.

Implicados clave de un proyecto:

- Director de proyecto.
- Cliente.
- Desarrolladores o empresa de desarrollo.
- Sponsor.

Gestionar las expectativas de los implicados suele ser difícil porque a menudo tienen diferentes objetivos, que pueden estar enfrentados o en conflicto. Los roles y las responsabilidades de los distintos implicados en el proyecto, la mayoría de las veces, se superponen.

En general las diferencias entre los implicados deberían resolverse a favor del cliente. No obstante las necesidades y expectativas de los demás implicados se verían perjudicadas. Buscar la solución apropiada a este conflicto de intereses es uno de los mayores retos dentro de la gestión y dirección de proyectos.

La gestión de recursos humanos incluye los procesos necesarios para hacer más efectivo el uso de las personas involucradas en el proyecto. Esta gestión incluye a todos los implicados en el proyecto (clientes, sponsors, etc.).

Los principales procesos incluyen:

- **Plan de organización y planificación.** Identificar, documentar y asignar roles y responsabilidades y relaciones de los implicados.
- **Adquisición del personal.**
- **Formación.**
- **Motivación.**
- **Equipo de desarrollo y estructura del equipo.**

Estos procesos interactúan unos con otros y con los procesos de las otras áreas de conocimiento, e implican la consecución de muchas tareas, algunas de estas tareas incluyen:

- Liderazgo, comunicación, negociación, . . .
- Delegación, motivación, tutorización y otras relativas a las relaciones entre los individuos.
- Construcción del equipo, resolución de conflictos, y otras relativas al grupo.
- Adquisición, retención, relaciones laborales, normas de higiene y seguridad, . . .

El director y el equipo de gestión deben estar familiarizados con todas ellas, y tener claro cómo deben aplicar todo ello al proyecto. Por ejemplo:

- Debido a la naturaleza temporal de los proyectos, el personal y las relaciones entre los implicados serán por lo general de carácter temporal y nuevas. El equipo de gestión del proyecto debe ser cuidadoso al seleccionar las técnicas adecuadas para cada una de las relaciones temporales que se establezcan.
- La naturaleza y número de los implicados en el proyecto a menudo cambia cuando el proyecto cambia de fase en su ciclo de vida. Las técnicas efectivas en una fase no tienen porque serlo en otra. El equipo de gestión debe ser cuidadoso con el uso de las técnicas apropiadas a las necesidades actuales del proyecto.
- Las actividades administrativas de los recursos humanos raramente son responsabilidad del equipo de gestión del proyecto. No obstante el equipo debe tener un conocimiento de los requisitos administrativos que se revele como suficiente para asegurar una consecución exitosa del proyecto.

Sabemos con seguridad que los temas relacionados con personas tienen un mayor impacto en la productividad del software y por tanto en la calidad del mismo. Estudio tras estudio se ha descubierto que la productividad de programadores concretos de similar nivel de experiencia puede variar en un factor de la menos 10 a 1 [Vallet y McGarry, 1989].

Los estudios también han descubierto variaciones en la eficiencia de equipos completos del orden de 3, 4, o 5 a 1 [Boehm, 1981]. Después de 20 años de experiencia en proyectos reales, los investigadores del Laboratorio de Ingeniería del Software de la NASA han llegado a la conclusión de que la tecnología no es la respuesta; los métodos más efectivos son aquellos que sacan partido al potencial humano de sus desarrolladores [Basili y McGarry, 1995].

Está claro que todo lo relacionado con *peopleware* (disciplina que experimenta sobre temas relacionados con el personal) influye fuertemente en la productividad, ahora también queda muy claro que cualquier organización que trate seriamente de mejorar la productividad primero debe ocuparse de temas relacionados con personal, como la motivación, equipo de trabajo, selección de personal y formación. Hay otras formas de mejorar la productividad, pero la gestión de personal ofrece los mayores beneficios potenciales.

En lo que a selección se refiere Barry Boehm presenta cinco principios para la selección de personal [Boehm, 1981]:

- **Máximo talento.** Usar poco y buen personal.
- **Trabajo adecuado.** Asignar tareas según la habilidad y motivación de la gente disponible.
- **Progresión profesional.** Ayudar a la gente a actualizarse por sí misma en vez de obligarles a trabajar donde más experiencia tienen o donde son más necesarios.
- **Equilibrado del equipo.** Seleccionar a gente que se complemente y armonice con los demás.
- **Eliminar la inadaptación.** Eliminar y reemplazar a los miembros problemáticos del equipo lo antes posible.

Otros factores que pueden marcar la diferencia son la habilidad de diseño del personal, la habilidad en programación, la experiencia en el entorno y la maquina y la experiencia en el área de aplicación.

La forma de organizar al personal tiene un gran efecto sobre la eficiencia con la que trabajen. Hay que sacar partido a la estructura de los equipos para que concuerden con el tamaño del proyecto, las características del producto y los objetivos de planificación. Un proyecto software específico también puede sacar provecho de la especialización apropiada.

Una persona que carece de motivación no va a querer trabajar duro, y prefiere dejarse llevar. La motivación es el único factor que provocará que una persona renuncie a las tardes y a los fines de semana sin que se le pida. Pocos otros factores pueden aplicarse a tanta gente dentro de tantos equipos en tantas empresas. La motivación es potencialmente el aliado más fuerte que tenemos.

La gestión de los recursos humanos ya se ha revelado como una parte decisiva en todo proyecto, combinando esta gestión con la de las otras áreas y con unos buenos métodos, debería ser suficiente. Pero no es así, uno de los motores que han hecho posible que este documento viera la luz, fue el descubrimiento de que una gran parte de los errores en los que se cae en la gestión de proyectos son por causas directamente relacionadas con la mala gestión de los recursos humanos. Por muy bien hecha que supongamos esta una planificación o eficaz que creamos nuestra gestión, aún podemos cometer un error que de al traste con la productividad.

Según Mc Connell [1996] podemos caer en lo que denomina “errores clásicos” que él viene a definir como el uso de técnicas de desarrollo infectivas que han sido elegidas con tanta frecuencia, por tanta gente, con resultados tan malos y predecibles que son dignas de tal denominación. Los desarrolladores, directivos y clientes normalmente tienen buenas razones para tomar las decisiones que toman y la apariencia seductora de los errores clásicos es una de las razones de que esos errores se comentan tan a menudo, pero debido a que se han cometido muchas veces, sus consecuencias se han hecho fáciles de predecir. Y los errores clásicos rara vez producen los resultados que la gente espera.

El común denominador de esos errores es que un buen y rápido desarrollo no se consigue si se evitan, pero si no se evitan, seguro que se consigue un desarrollo lento y malo.

La lista de errores esta dividida en cuatro dimensiones:

Personas.

1. Motivación débil. Estudio tras estudio se ha demostrado que la motivación probablemente tiene mayor efecto sobre la productividad y la calidad que ningún otro factor [Boehm, 1981].

2. Personal mediocre. La capacidad individual de los miembros del equipo, así como sus relaciones como equipo, probablemente tienen la mayor influencia en la productividad [Boehm, 1981].

3. Empleados problemáticos incontrolados. Un fallo al tratar con personal problemático amenaza la velocidad de desarrollo. Un fallo en una decisión al tratar con un empleado problemático es una de las quejas más comunes que tienen los miembros del equipo respecto de sus responsables (Larson y LaFasto, 1989).

4. Hazañas. El énfasis en los comportamientos heroicos fomenta correr un riesgo extremo e impide la cooperación entre los múltiples elementos que contribuyen al proceso de desarrollo del software.

Algunos directivos fomentan el comportamiento heroico cuando se concentran con demasiada firmeza en actitudes “ser capaz de”. Elevando estas actitudes por encima de informes del estado exactos y a veces pesimistas, los directivos de estos proyectos, coartan su capacidad de tomar medidas correctivas. Ni siquiera saben que tienen que emprender acciones correctoras hasta que el daño ya esta hecho.

5. Añadir más personal a un proyecto retrasado. Cuando un proyecto se alarga, añadir más gente puede quitar más productividad a los miembros del equipo existente de la que añaden los nuevos miembros.

6. Oficinas repletas y ruidosas. La mayoría de los desarrolladores consideran sus condiciones de trabajo como insatisfactorias. Los trabajadores que están en oficinas silenciosas y privadas tienden a funcionar significativamente mejor que aquellos que ocupan cubículos en salas ruidosas y repletas. Los entornos repletos y ruidosos alargan los planes de desarrollo.

7. Fricciones entre los clientes y los desarrolladores. El principal efecto de esta fricción es la mala comunicación, y los efectos secundarios de la mala comunicación incluyen el pobre entendimiento de los requerimientos, pobre diseño de la interfaz de usuario y, en el peor caso, el rechazo del cliente a aceptar el producto acabado. En la mayoría de los casos suelen ser conflictos de personalidad entre dos grupos.

8. Expectativas poco realistas. Aunque por si mismas las expectativas irreales no alargaran el plan, contribuyen a la percepción de que el plan de desarrollo es demasiado largo, y de que puede ser malo.

9. Falta de un promotor efectivo del proyecto. Sin un promotor ejecutivo efectivo, el resto de personal de alto nivel de a empresa puede forzar a que se acepten fechas de entrega irreales o hacer cambios que debiliten el proyecto.

10. Falta de participación de los implicados. Todos los principales participantes del esfuerzo de desarrollo de software deben implicarse en el proyecto. La cooperación estrecha solo se produce si se han implicado todos los participantes, permitiendo una coordinación precisa del esfuerzo para un desarrollo rápido y satisfactorio, que es imposible conseguir sin una buena participación.

11. Falta de participación del usuario. Los proyectos que no implican al usuario desde el principio corren el riesgo de que no se comprendan los requerimientos del proyecto, y son vulnerables a que se consuma tiempo en prestaciones que más tarde retrasaran el proyecto.

12. Política antes que desarrollo. Primar la política en vez de los resultados es fatal para el desarrollo.

13. Ilusiones. Las ilusiones no son solo optimismo. Realmente consisten en cerrar los ojos y que todo funcione cuando no se tienen las bases razonables para pensar que será así. Las ilusiones al comienzo del proyecto llevan a grandes explosiones al final. Impiden llevar a cabo una planificación coherente y pueden ser la raíz de más problemas en el software que todas las otras causas combinadas.

En lo referente a los errores clásicos en el proceso, el producto y la tecnología, nos limitaremos a una enumeración, excepto cuando se establezca una relación clara con los recursos humanos.

Proceso.

Los errores relacionados con el proceso ralentizan los proyectos porque malgastan el talento y el esfuerzo del personal.

14. **Planificación excesivamente optimista** predispone a que el proyecto falle por infravalorar el alcance del proyecto, minando la planificación efectiva, y reduciendo las actividades críticas para el desarrollo, como el análisis de requerimientos o el diseño. También supone una presión excesiva para los desarrolladores, quienes a largo plazo se ven afectados en su moral y en su productividad.

15. **Gestión de riesgos insuficiente.**

16. **Fallos de los contratados externos.**

17. **Planificación insuficiente.**

18. **Abandono de la planificación bajo presión.** Los equipos de desarrollo hacen planes y rutinariamente los abandonan cuando se tropiezan con un problema en la planificación [Humphrey, 1989]. El problema no está en el abandono del plan, sino más bien en fallar al no crear un plan alternativo, y caer entonces en el modo de trabajo de codificar y corregir.

19. **Pérdida de tiempo en el inicio difuso.**

20. **Escatimar en las actividades iniciales.**

21. **Diseño inadecuado.**

22. **Escatimar en el control de calidad.**

23. **Control insuficiente de la directiva.**

24. **Convergencia prematura o excesivamente frecuente.**

25. **Omitir tareas necesarias en la estimación.**

26. **Planificar ponerse al día más adelante.**

27. **Programación a destajo.**

Producto.

28. **Exceso de requerimientos.**

29. **Cambio de las prestaciones.**

30. **Desarrolladores meticulosos.** Los desarrolladores encuentran fascinante la nueva tecnología y a veces están ansiosos por probar nuevas prestaciones de su lenguaje o entorno, o por crear su propia implementación de una utilidad bonita que han visto en otro producto, la necesite o no su producto. El esfuerzo requerido para diseñar, implementar, probar, documentar, o mantener estas prestaciones innecesarias alarga el plan.

31. **Tiras y aflojas en la negociación.**

32. **Desarrollo orientado a la investigación.**

Tecnología.

33. **Síndrome de la panacea.**

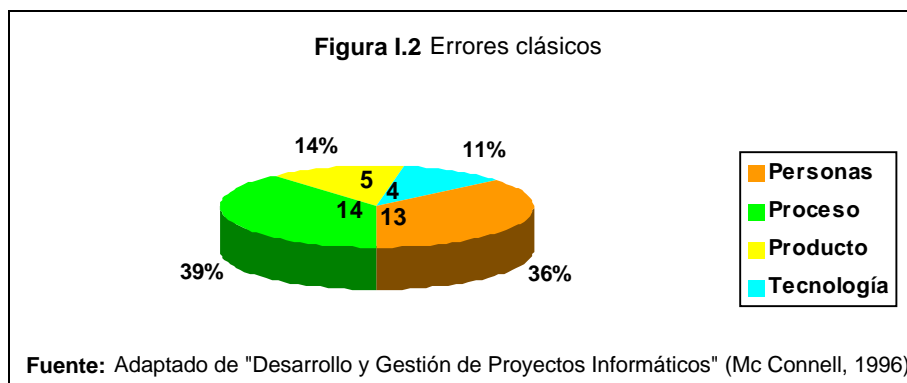
34. **Sobreestimación de las ventajas de nuevas herramientas o métodos.**

35. **Cambiar de herramientas a mitad del proyecto.**

36. **Falta de control automático del código fuente.**

Los errores clásicos son muchos más, los que se indican son los más comunes y los más serios.

De los 36 errores clásicos, 13 de ellos son directamente causados por una mala gestión de los recursos humanos, lo que significa el 36% del total, tan solo superados por los 14 de producto (figura I.2) y 3 errores clásicos de la dimensión de producto están relacionados con una mala gestión del personal o desembocarán en problemas que nos harán caer en alguno de los 13 errores clásicos relacionados con las personas. Sin duda todos los errores clásicos de las dimensiones de proceso, producto y tecnología tendrán una repercusión más o menos grave sobre los recursos humanos de nuestro proyecto y la planificación del mismo, lo que al final, con casi total seguridad se habrá hecho una mala gestión.



Todo ello nos muestra una vez más lo necesaria que una gestión eficiente de los recursos humanos, y la importancia y peso específico de dicha gestión en el conjunto de un proyecto informático. Una mala gestión de los recursos humano será la diferencia entre el éxito y el fracaso de nuestro proyecto, por muy buena que sea la planificación y eficiente la gestión en las otras áreas.

Este documento seguirá lo que parece la trayectoria más lógica, una vez hecha la introducción general (y emocional), y situada la gestión de los recursos humanos en un proyecto y dentro del Project Management en general, viajado en el "ciclo de vida" de la gestión de los recursos humanos en proyectos informáticos, el que la lógica parece dictar comienza con la selección, debería seguir con la formación y la política de empresa para ya entrar en lo que sería la gestión una vez comenzado el proyecto: planificación, motivación, equipo de trabajo, estructura del equipo.

Se incluyen una serie de métodos relacionados o con gran incidencia sobre los recursos humanos y la gestión de los mismos, recomendables para un buen desarrollo: Entornos Productivos, Compromiso (Signing Up), Gestión Theory-W y Horas Extras Voluntarias.

Y para finalizar una comparación con un método, "*el método*" que tan extendido está y que tantas contraindicaciones tiene, presión, más presión, trabajo a destajo, etcétera. Rebatiremos brevemente con los conocimientos ya adquiridos, lo cáustico de las prácticas habituales (y dañinas) en la gestión de proyectos informáticos.

2.- Selección, Formación, Dirección

Aunque la selección y la formación, se podría considerar como tareas separadas a la gestión de los recursos humanos, el éxito o fracaso de las mismas afectará a nuestro proyecto casi en la misma medida que una buena o mala gestión del personal, una expresión que define muy bien este fenómeno es: *“luego, pasa lo que pasa”*. La participación directa de los equipos de dirección y gestión de proyectos es de suma importancia para el éxito en la adquisición de personal y asegura que los perfiles de los candidatos y las personas seleccionadas, se ajustan y tienen las características necesarias el proyecto y por ende el buen desarrollo del mismo.

2.1.- Selección

En el proceso de incorporación de los futuros colaboradores a la empresa, existen dos momentos fundamentales: la convocatoria y la selección. El planteamiento del primero condiciona el desarrollo y el éxito del segundo.

La convocatoria es la fase en que la empresa debe buscar y obtener los candidatos suficientes, que reúnan las características y condiciones idóneas.

La selección se ocupa de conocer, estudiar y evaluar a los candidatos, para determinar los que poseen las características requeridas a un nivel determinado, que les permitan desempeñar las tareas designadas con mayor probabilidad de satisfacción.

La elección de las mejores personas es un punto de partida básico para garantizar el éxito de un proyecto.

Unos procesos de convocatoria y selección rigurosos pueden eliminar consecuencias negativas como la mala calidad del producto, un bajo rendimiento y la falta de motivación.

En cualquier caso, y dada la transcendencia de las decisiones tanto para el momento actual como para el desarrollo futuro, plantear correctamente el proceso de convocatoria y la selección es una labor esencial.

2.1.1.- EL PROCESO DE CONVOCATORIA

El objetivo del proceso de convocatoria es obtener el suficiente número de personas capaces, interesadas en la oferta de trabajo realizada, para poder llevar a cabo un posterior proceso de selección sin limitaciones, que posibilite determinar el candidato más adecuado.

La necesidad surge como consecuencia de la existencia de un puesto que por distintos motivos ha quedado vacante.

Para poder saber qué tipo de persona es la más idónea se hace necesario contestar a las dos cuestiones siguientes:

- Que contenido tiene el puesto (Descripción del puesto).
- Cuáles son las características personales y profesionales que debe reunir su ocupante (Perfil del puesto).

Descripción del puesto de trabajo. El análisis y descripción del puesto de trabajo permite saber:

- Como se denomina cada puesto.
- Dónde se ubica dentro de la estructura organizativa general.
- Qué se hace en él y para qué.
- Dónde, cómo y cuándo actúa.

Toda esta información sistematizada se recoge en fichas descriptivas que, como norma general, incluyen datos referentes a los siguientes aspectos:

Ficha descriptiva del puesto

- Nombre del puesto.
- Área, Departamento o Sección a la que pertenece.
- Dependencia jerárquica y puestos que dependen de él.
- Función básica o principal del puesto.
- Funciones o actividades específicas de las que debe ocuparse.
- Responsabilidades:
 - Alcance y nivel de las decisiones que debe tomar.
 - Implicaciones que tiene sobre aspectos económicos.
 - Grado, momento y carácter de la supervisión que la empresa ejerce sobre el puesto.
- Relaciones internas y externas que se mantienen en el puesto, así como su frecuencia, nivel y repercusiones.
- Medios de apoyo técnico que utiliza (información, equipos, etc.).
- Condiciones especiales (viajes, traslados, dedicación, etc.).
- Condiciones ambientales (riesgo, entorno físico, etc.).

Fuente: [INEM, 1991].

Los datos de la ficha anterior permiten valorar objetivamente el puesto y sirven de base para definirlos requisitos de la persona que lo desempeñe y los niveles retributivos de dicho puesto.

Perfil del puesto. De acuerdo a las características del puesto podemos determinar las condiciones que deben reunir los aspirantes. El conjunto de dichas condiciones se denomina *perfil*.

Perfil o Profesiograma del puesto

- Requisitos personales: Edad, sexo, etc.
- Conocimientos: Tipo y nivel académico o de formación profesional.
- Requisitos profesionales: Grado y clase de experiencia necesaria.
- Requisitos mentales o aptitudes: Relativa a las habilidades en el desempeño del puesto (verbales, numéricas, espaciales, atención, inteligencia general, etc.).
- Requisitos físicos: Aspecto, salud, etc.
- Requisitos de personalidad y motivación: Carácter, expectativas, capacidad de trabajo, etc.

Fuente: [INEM, 1991].

Una vez definido el perfil, hay que reflexionar sobre las posibilidades de encontrar personas adecuadas al mismo. Si después de ello se determina que la previsión de respuesta es suficiente hay que establecer la forma en la que se va a llevar a cabo el proceso de adquisición, esto es:

- ¿Dónde se encuentran las personas que se adecuan al perfil?
- ¿Cómo hacerles llegar la información y que se interesen por la oferta?

Donde. Las fuentes de adquisición pueden ser tanto internas como externas. Las fuentes internas pueden ser: *Planes de rotación-traslado o planes de promoción*. En cuanto a las externas son muy numerosas, las principales son: *Solicitudes y ofrecimientos, Oficinas de empleo, Centros de formación, Asociaciones, Empresas de la competencia, Empresas especializadas y consultoras de recursos humanos*.

Cómo. La oferta puede darse a conocer de forma directa o a través de medios de comunicación.

2.1.2.- ETAPA DE PRESELECCIÓN

Durante la semana siguiente a la difusión de la oferta se reciben la mayoría de las respuestas. Tras repasar la descripción y el perfil del puesto, se pueden establecer tres grupos según su adecuación a las exigencias:

1. Candidatos que cubren en mayor medida el perfil (pasarán al proceso de selección).
2. Los que no cubren con entera satisfacción alguno de los requisitos básicos (casos dudosos, se reservarán).
3. El resto de los candidatos, que claramente no se adecuan al perfil (se descartan).

Realizar una buena preselección puede favorecer en gran medida el éxito de la selección, actuar con detenimiento en esta etapa es fundamental para aumentar la fiabilidad y agilidad de la siguiente fase.

2.1.3.- EL PROCESO DE SELECCIÓN

El objetivo de la selección es conocer a los candidatos, analizar y comprobar la información recogida y completarla con datos sobre sus experiencias anteriores y sus expectativas y motivaciones.

La información que debe obtenerse es la siguiente:

- Historial Académico.
- Historial profesional.
- Conocimientos profesionales específicos.
- Inteligencia y aptitudes.
- Personalidad y Motivaciones.

La entrevista de selección. Es una parte fundamental del proceso de selección, se pretende conseguir una visión amplia de los comportamientos y resultados pasados del candidato. Es una técnica valiosa si se planifica y encamina correctamente hacia los objetivos que se persiguen.

La calidad y resultados de la entrevista dependen en alto grado de las preguntas que se formulen. Las preguntas abiertas, las que permiten al entrevistado contestar con iniciativa propia son las más eficaces. Las preguntas cerradas (“SI” o “NO”) no aportan demasiada información.

Es adecuado hacer preguntas sobre hechos reales que hagan referencia a situaciones vividas por el entrevistado. Sin embargo, cuando se trata de puestos de toma de decisiones son válidas preguntas teóricas o hipotéticas (¿Qué haría si ...?). Hay que evitar a toda costa preguntas que sugieren la respuesta, el entrevistado tiende a adecuar la respuesta a la imagen que el entrevistador le transmite. Sólo aportan confusión y datos falseados.

Los candidatos pueden aportar otro tipo de información no verbal como gestos, expresiones, movimientos, tono de voz, etc. que pueden indicar inseguridad, tensión, inexperiencia, etc. Deben ser también recogidos.

Las entrevistas deben tener una estructura flexible que contemplaría las siguientes fases:

- **Toma de contacto.** Preguntas sencillas y poco trascendentes para relajar la tensión.
- **Preguntas.** Repaso a las actividades y responsabilidades anteriores.
- **Información al candidato.** Sobre el puesto, la empresa, condiciones económicas y de contratación, etc.
- **Ofrecer al candidato la oportunidad de preguntar.** Aportan información adicional sobre la comprensión del puesto y su motivación.
- **Cierre de la entrevista.** Aclaraciones, forma de contacto, despedida.

Los test psicológicos. Son instrumentos de medida de capacidades y características humanas. Poseen valor de diagnóstico y predicción. Se utilizan para obtener la situación psicológica del individuo o evaluar su potencial de desarrollo.

Los test pueden clasificarse con respecto a los aspectos que evalúan en:

- **De inteligencia.** Miden la capacidad mental, análisis, deducción, etc.
- **De aptitudes.** Miden habilidades verbales, numéricas, espaciales, etc.
- **De personalidad.** Miden factores y rasgos de carácter.
- **Cuestionarios de intereses.** Miden aspectos como la responsabilidad, aceptación de normas, etc.

En general, los test son instrumentos que deben ser interpretados y valorados por especialistas en la conducta humana.

Prueba y exámenes profesionales. Buscan una valoración objetiva de la capacidad real de los candidatos para desempeñar la labor del puesto al que optan. La ventaja es poder presentar cuestiones o casos específicos para cada ocasión.

2.1.4.- TOMA DE DECISIÓN E INCORPORACIÓN

Una vez finalizado el proceso de selección comienza la tramitación e integración del nuevo colaborador. La presentación correcta de la empresa, de sus componentes, de las actividades de deberá realizar y de los objetivos que se esperan de él. Son cuestiones esenciales para una integración satisfactoria del seleccionado en el proyecto.

2.2.- Formación

Las necesidades de formación son constantes. Hay que capacitar a los empleados que no están obteniendo un rendimiento suficiente.

Habría que averiguar si las dificultades de ejecución se pueden solucionar con formación o es preciso buscar otra solución.

Por otra parte la introducción de nuevos equipos, métodos o procedimientos requerirán también nuevas acciones de formación. La formación ayuda a adelantarse a los cambios.

La formación es importante, pero no es el caso, por ello solo comentaremos la parte que más de lleno podría recaer sobre un director o un equipo de gestión, la detección de necesidades de formación.

Hay que ser escuetos, pero no está de más un breve comentario a modo de visión general.

2.2.1.- ESTIMACIÓN DE NECESIDADES

Cuando el rendimiento está por debajo del requerido, diremos que nos encontramos ante una *necesidad de ejecución*, lo que puede indicar:

- Una necesidad de formación.
- Una necesidad de organización.
- Ambas.

Síntomas. La valoración individual y de grupo que realiza el mando sobre la ejecución de las distintas tareas, es el mejor indicador posible.

Algunos síntomas podrían ser: baja producción, aumento del tiempo de desarrollo, baja disponibilidad, absentismo, etc.

Causas. Las dificultades de ejecución pueden ser debidas a factores asociados con el individuo, la organización, ambos.

Si se puede hay que evaluar el impacto coste-beneficio que supone la eliminación de la dificultad, esto es de gran utilidad para justificar la inversión en formación.

Los factores debidos al individuo estarán en relación con el modo de ejecutar su trabajo.

Los factores debidos a la organización pueden ser por un planteamiento equivocado de los requisitos de ejecución o por una recompensa, sin saberlo, del comportamiento inadecuado.

Identificación de soluciones y las dificultades de ejecución. Analizadas las causas, se pueden identificar las soluciones. Sólo algunas soluciones tienen que ver con la formación habitual, en la tabla de la página siguiente se describen algunas de las más comunes:

<u>SITUACION</u>	<u>SOLUCIONES</u>
1.- La persona no reúne los requisitos previos para aprender el trabajo	1.- Impartir los requisitos previos.
2.- La persona reúne los requisitos previos, pero no tiene la capacidad requerida	2.1 Proporcionar formación formal 2.2 Proporcionar formación en el puesto de trabajo 2.3 Decir o demostrar la manera de hacer el trabajo (sólo en caso de tareas sencillas, fáciles de aprender).
3.- La persona no cuenta con la capacidad requerida ni con los requisitos previos, y probablemente no es capaz de tenerlos.	3.1 Reestructurar el puesto, adecuarlo a sus capacidades. 3.2 Transferir la persona a otro puesto.
4.- La persona ha realizado bien el trabajo en el pasado. En el presente realiza con poca frecuencia las tareas que conllevan dificultades de ejecución.	4.1 Programar una práctica frecuente de esas tareas. 4.2 Formarle en relación con dichas tareas. 4.3 Suministrar un manual de referencia o material autodidáctico en relación con esas tareas. 4.4 Proporcionar ayuda/s para el trabajo (listas de comprobación, diagramas, etc.)
5.- La persona actuó correctamente en el pasado, y ahora realiza dichas tareas frecuentemente.	5.1 Ampliar y enriquecer el trabajo. 5.2 Facilitarle frecuente información de cómo realiza el trabajo.
6.- La persona tiene las capacidades y realiza con frecuencia las tareas en cuestión, pero no es recompensado suficientemente por el trabajo bien hecho.	6.1 Instituir o incrementar los premios a la buena ejecución.
7.- La buena labor es "castigada" inconscientemente por el mando.	7.1 Suprimir las consecuencias que son vistas como "castigo" de una ejecución correcta.
8.- El trabajo deficiente es inconscientemente gratificado.	8.1 Suprimir el "premio" a la ejecución deficiente.
9.- La realización deficiente del trabajo se tolera	9.1 Incrementar o imponer consecuencias negativas por ejecución deficiente (suprimir aumento de sueldo, ascensos, etc)
10.- La realización deficiente es consecuencia de la organización del trabajo	10.1 Replanificar u organizar los métodos o la secuencia del trabajo. 10.2 Informar al empleado sobre lo que se espera de él (descripción de su puesto de trabajo, cometido, objetivos) 10.3 Suprimir las demandas que entren en conflicto con su tiempo.
11.- La ejecución deficiente está producida por factores del entorno de la organización.	11.1 Facilitar herramientas y/o medios adecuados. 11.2 Reorganizar, replanificar el puesto y/o el centro de trabajo (iluminación, disposición, colores, ruido excesivo, etc.)
12.- La ejecución incorrecta viene determinada por el reglamento y/o estructura de la organización.	12.1 Cambiar las políticas de la empresa que impiden la ejecución satisfactoria. 12.2 Reorganizar la estructura administrativa de la empresa.

Fuente: [INEM, 1991].

2.3.- Dirección

El fracaso o el éxito de un proyecto dependerá de la calidad de la gente que en él trabaje. El centrarse en la técnica distorsiona la visión de lo que está sucediendo en el proyecto. Casi nunca un proyecto fracasa porque la gente no sabe cómo usar las técnicas más avanzadas, pero muchos proyectos se deshicieron porque los directivos emitieron órdenes inadecuadas o porque la falta de liderazgo lo llevó al fracaso.

Actualmente se reconoce que las personas son la clave para el éxito. Esto se refleja en el abandono de las jerarquías impersonales que tanto dominaron el management antaño.

Hasta no hace mucho, la opinión generalizada era que la tarea de los ejecutivos consistía en dirigir y controlar; hoy, en cambio, se sostiene que la función clave de los gerentes es *apoyar*, es decir, crear un ambiente que permita a los empleados trabajar lo mejor posible. Se llegó a esta situación merced a dos influencias: la obsesión por la satisfacción del cliente y la creciente complejidad del mundo actual. Pronto se vio que las estructuras tradicionales, de jerarquía vertical, no podían satisfacer las expectativas de los clientes.

El alto nivel de complejidad del mundo actual ha contribuido también a que los managers *apoyen* a sus empleados, en vez de *dirigirlos*. En este mundo complejo, no es posible que *una sola persona* posea la competencia suficiente para encarar *todos los problemas*.

Esta complejidad exige que las decisiones sean tomadas por equipos transfuncionales que tengan el conocimiento colectivo necesario para tomar decisiones sobre asuntos importantes.

La dirección de proyectos tradicional fue siempre vertical, pero, en proyectos menos predecibles, propios de la era de la información, la toma de decisiones debe repartirse entre los miembros del equipo. Hay demasiada especialización, complejidad e incertidumbre para que una sola persona tome las decisiones, como si tuviera todas las respuestas.

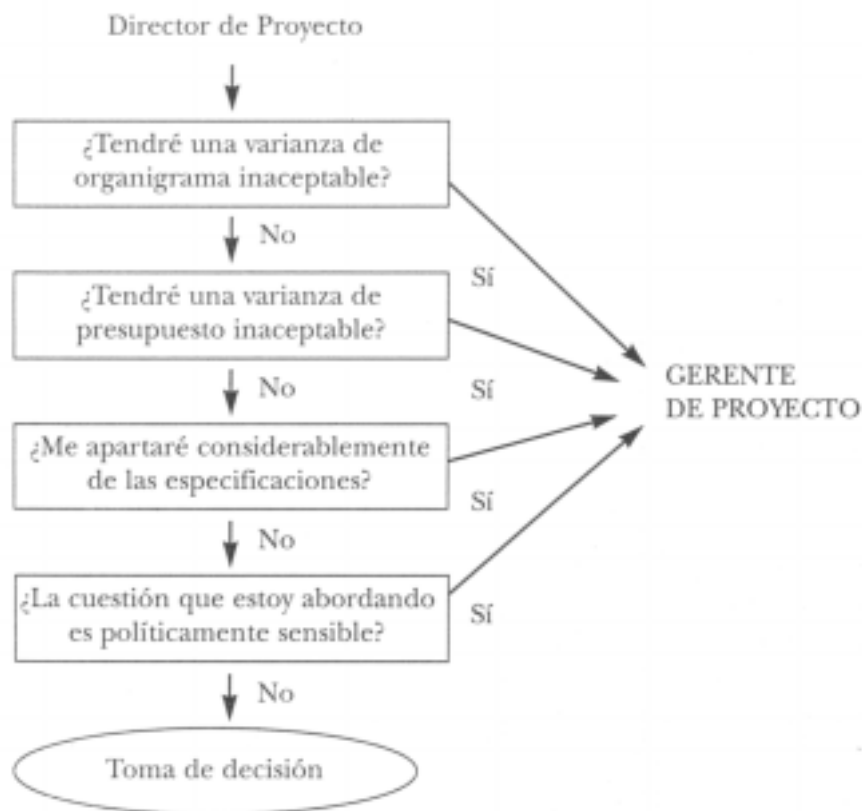
2.3.1.- ¿QUIÉN MANDA AQUÍ?

Una de las preguntas más difíciles de responder es: ¿Y aquí quién manda?.

Las decisiones importantes son tomadas por diversas personas a lo largo de todo el proceso y "Hasta cierto punto, aquí *todos mandan*".

Desde luego, la dispersión de la toma de decisiones a lo largo del proyecto puede producir confusión y conflictos, pero cuando un proyecto es extremadamente complejo para ser manejado en un estilo rígidamente jerárquico, la toma de decisiones se distribuye a lo largo del proceso de realización de un proyecto, con una cadena de mandos claramente establecida.

La figura 2.1 ilustra este punto. Se ve allí que los directores de proyecto son a menudo "managers por excepción". Ello quiere decir que se le da al equipo amplia libertad de acción, cuando es evidente la carencia para la toma de decisión o cambia el rumbo fijado para el proyecto, el director del proyecto interviene directamente en la toma de decisiones.

Figura 2.1. Management por excepción.

Fuente: [Davison, 1999]

2.3.2.- RESPONSABILIDADES DEL DIRECTOR DE PROYECTO

Si a un director de proyecto le preguntan cuáles son sus responsabilidades, es muy probable que conteste: “Hacer el trabajo a tiempo, dentro del presupuesto y según las especificaciones”, sin lugar a dudas sus responsabilidades van más allá. Debe, también contribuir al desarrollo del personal, actuar como intermediario entre los directivos y el equipo de desarrollo, y transmitir a la organización todo lo que han aprendido en el proceso.

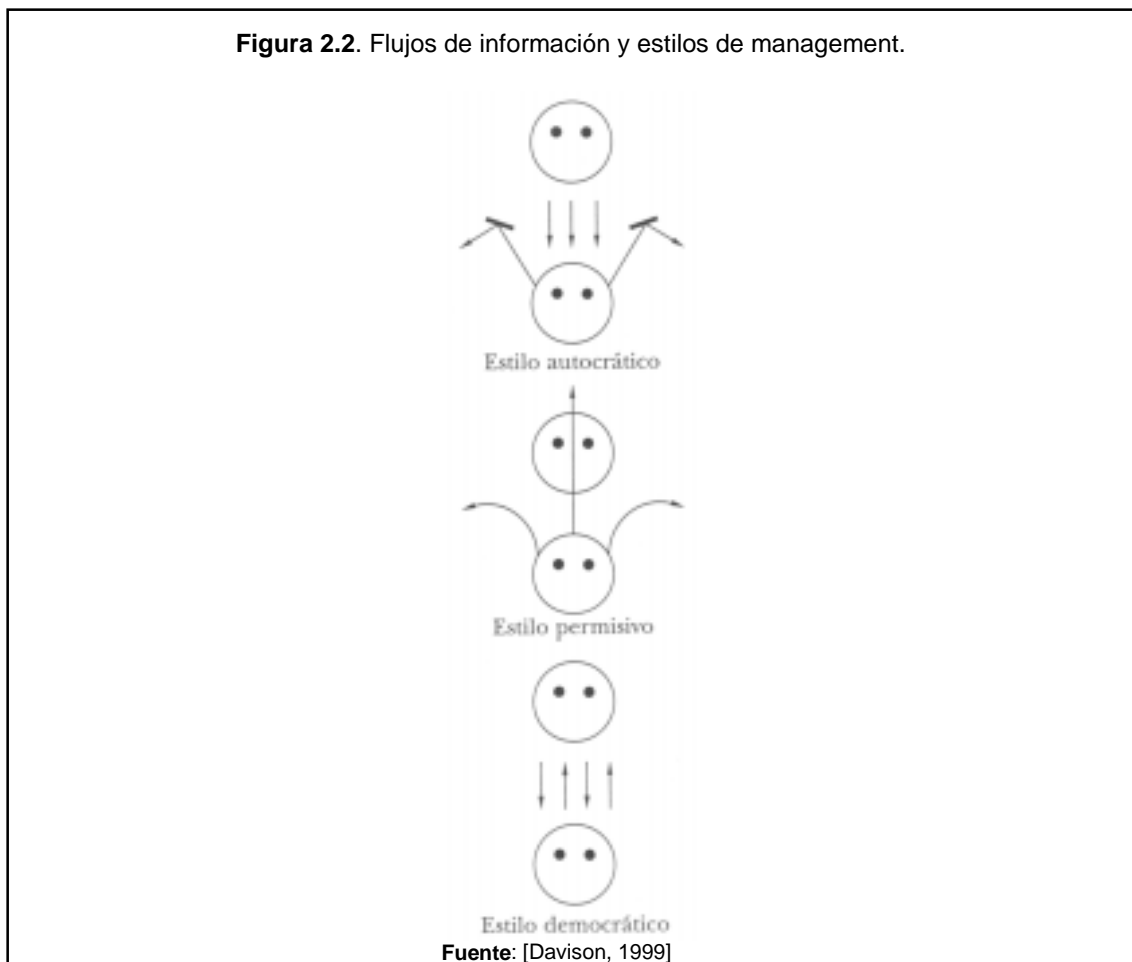
Los gerentes de proyecto tienen por encima a los altos niveles del management y por debajo a los desarrolladores. Su situación es delicada, forman parte del plantel ejecutivo y se espera de ellos que obren en consecuencia. Son el canal a través del cual fluye la información desde los directivos hacia los empleados. A través de ellos el equipo de trabajo del proyecto puede atisbar los objetivos organizacionales y los deseos de la dirección. Lamentablemente, los directores de proyecto corren el riesgo de que se los identifique como aliados incondicionales de los directivos de la firma.

Por otra parte, los gerentes de proyecto forman parte del plantel de empleados y les brindan a los ejecutivos una visión de las necesidades, capacidades y deseos de los empleados. Y al desempeñar este papel tampoco deben dar la impresión de ser “representantes” de los desarrolladores.

2.3.3.- LOS ESTILOS DE DIRECCIÓN

Un estilo de dirección es la manera en que los gerentes interactúan con su equipo. Hay tres estilos básicos que figuran en la bibliografía más usual de management: el autocrático, el permisivo y el democrático.

Para comprender la dinámica de estos tres estilos y apreciar sus diferencias es conveniente analizarlos en función de los flujos de información (figura 2.2).



Estilo autocrático. A estos directores de proyecto no les interesa procesar la información que viene de afuera ni tampoco la realimentación proveniente del equipo. A veces sustentan una política de puertas abiertas, pero sólo en la medida en que buscan la información del exterior, ya que después no hacen nada con ella, son autocráticos.

Ventajas: es adecuado para los proyectos de rutina y bajo riesgo, en los que el equipo se limita a llevar adelante el plan como fue establecido, en este caso, La realimentación del equipo no es tan decisiva como en un proyecto fluido y de alto riesgo. Este método es eficaz cuando es preciso tomar decisiones rápidamente.

Desventajas. Lleva a la desmoralización del equipo, ya que los desarrolladores no pueden hacer ninguna contribución al proceso de toma de decisiones. Puede llevar a tomar decisiones erróneas, ya que el director no recibe suficiente información desde el exterior.

El *laissez-faire*: estilo permisivo. La toma de decisiones en un ambiente permisivo es muy difusa, en francés, *laissez-faire* significa “dejar hacer”. En estos casos hay un escaso flujo de información o un flujo aleatorio, que no se canaliza adecuadamente.

En un sistema permisivo, el equipo del proyecto debe ser capaz de realimentar al equipo de gestión; pero lamentablemente los gestores no suelen actuar bien respecto de esa realimentación. En consecuencia, nos encontramos con que, a pesar de ser diametralmente opuestos, el método autocrático y el permisivo comparten un rasgo común: en ninguno de los dos la información fluye desde el equipo de trabajo hacia los directores del proyecto.

Este método puede ser eficaz en proyectos innovadores, en los que se fomenta la creatividad. Tal libertad de acción suele levantar la moral del equipo de desarrollo.

Desventajas. Puede llevar a la desorganización por falta de dirección y resulta simplemente desastroso cuando es preciso una rápida toma de decisiones.

Estilo democrático. Antes de tomar decisiones, los gerentes democráticos tratan activamente de recibir información originada en el equipo. Y es muy probable que este estilo sea el más eficaz.

Ventajas. Facilita la toma de buenas decisiones, ya que refleja un amplio espectro de puntos de vista. Incrementa el compromiso del equipo, ya que el conjunto de sus miembros participó en el proceso decisorio.

Desventajas. La primera de ellas es algo que los científicos políticos llaman la tiranía de la mayoría. Esto se produce cuando en un sistema democrático siempre impone determinada mayoría, en detrimento de lo que pasa a ser una perpetua minoría. No pasara mucho tiempo antes de que los desplazados se sientan desalentados y desilusionados con la toma de decisiones. La segunda desventaja se pone en evidencia cuando se confeccionan encuestas con los “votantes” equivocados y, en consecuencia, se toman decisiones sobre la base de una información incorrecta y una tercera desventaja es que puede ser inútil cuando es preciso tomar decisiones rápidas.

En este método, además, hay que contemplar el factor cultural; difícilmente podrá ser aplicado en sitios en los que ni siquiera conocen la faceta humana y política de la democracia.

Cómo elegir un estilo de management. En las situaciones reales de proyecto no es posible ni aconsejable que los directores empleen el 100 por 100 del tiempo para un solo estilo. Cada director debe adaptar su estilo para reflejar las circunstancias a las que se enfrenta. Se puede adoptar una modalidad permisiva con un equipo más íntimo, durante la fase creativa de diseño de un proyecto y después utilizar un método democrático durante la fase más rutinaria de la implementación. También puede suceder que la única manera de obtener buenos resultados, en ciertos casos, sea comportarse de manera autocrática.

El estilo empleado por el director del proyecto puede llegar a tener una enorme influencia sobre los resultados. El secreto consiste en saber qué estilo aplicar según las circunstancias. Esa decisión depende, en gran medida, del sentido común del director y de su capacidad para evaluar correctamente las situaciones y de varios factores de su personalidad, puede suceder que un gerente, por su personalidad, sea radicalmente incapaz de adoptar un estilo diferente de su temperamento básico.

2.3.4.- DIRECTIVOS Y RESPONSABLES TÉCNICOS

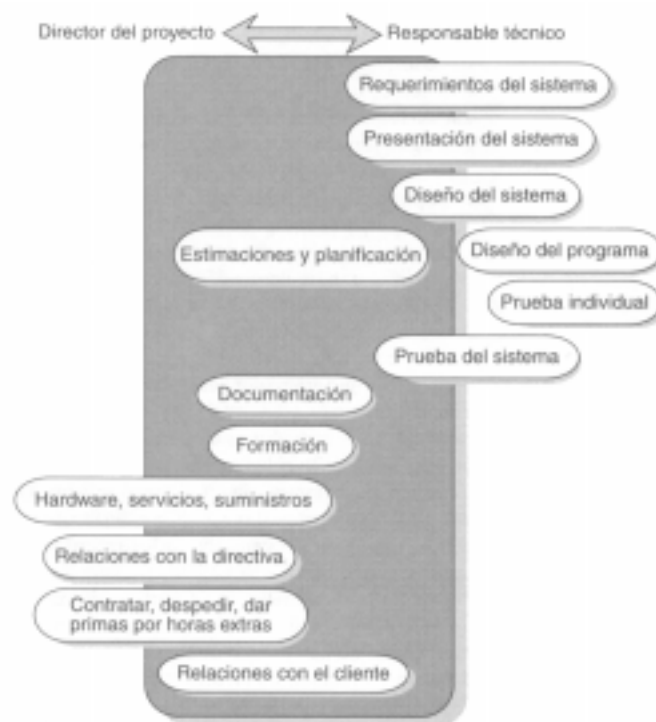
Los directivos y los responsables técnicos no trabajan siempre en una estrecha relación. Hay muchos problemas (superposición de responsabilidades, motivación, relaciones con los clientes, baja calidad, poca coincidencia en los objetivos, etc.) que pueden mejorarse cuando tienen una comunicación efectiva sobre los temas con los que están trabajando.

Uno de los mayores obstáculos para un rendimiento efectivo de la figura del responsable técnico es la falta de una división clara de responsabilidades entre el responsable técnico y el director.

En su forma más pura, el responsable técnico es el responsable del trabajo técnico, y es responsable de un solo equipo. El director es responsable de la dirección de los aspectos no técnicos del equipo, y suele ser responsable de dos o más proyectos. Desde el punto de vista del equipo, el papel del director consiste en liberar al responsable técnico gestionando ciertas tareas no técnicas. Desde el punto de vista de la organización, el papel del director es controlar al equipo para que se adapte a los objetivos de la organización. Algunos modelos de equipo (Capítulo 6), resultan mejores que otros para ayudar a mantener claras las diferencias entre ambas funciones. Como los detalles específicos de las relaciones entre responsable técnico son tan variables, resulta útil que ambos discutan sus funciones al principio del proyecto. Esto ayuda a evitar choques y vacíos de responsabilidad.

John Boddie [Boddie, 1987] publicó un interesante diagrama que describe su visión de la relación entre director y responsable técnico de un proyecto (figura 2.3). El diagrama aclara algunas cuestiones clave y se puede usar como punto de partida para debatir y aclarar las responsabilidades del director y del responsable técnico.

Figura 2.3. Responsabilidades del director y del responsable técnico.



Fuente: adaptado de [Boddie, 1987]

3.- La influencia de la Planificación

3.1.- Planificación excesivamente optimista

Aunque se podría pensar que la planificación imposible es un problema moderno, las planificaciones del software excesivamente optimistas son una tradición en el desarrollo del software. Capers Jones señala que la presión excesiva en la planificación es el más común de todos los problemas serios de ingeniería del software. “Las planificaciones excesivas o irracionales representan probablemente la influencia más destructiva de todas las que afectan al software” [Jones, 1994]. Trabajar bajo una presión extrema de la fecha límite se ha convertido en una tradición. La planificación agresiva impide una planificación precisa.

3.1.1.- CAUSAS FUNDAMENTALES DE LAS PLANIFICACIONES DEMASIADO OPTIMISTAS

Las causas fundamentales de las planificaciones demasiado optimistas son variadas. A continuación algunas de las más comunes:

- Hay un plazo límite externo e inmutable.
- Los responsables o clientes se niegan a aceptar un rango de estimación y hacen los planes basándose en una estimación puntual del “mejor caso”.
- Los responsables y desarrolladores subestiman deliberadamente el proyecto porque desean un incentivo o les gusta trabajar bajo presión.
- El proyecto es subestimado deliberadamente por la directiva u el responsable de ventas para proponer una oferta insuperable.
- Los desarrolladores subestiman un proyecto interesante para obtener fondos para trabajar en él. Esto es muy común sobre todo en proyectos internos.
- El responsable del proyecto es partidario de que los desarrolladores trabajarán más duro si la planificación es ambiciosa, y por tanto diseñan la planificación en consecuencia.
- La directiva principal o los clientes desean una fecha tope particular y el responsable del proyecto no puede contradecirlos.
- El proyecto comienza con una planificación realista, pero se añaden nuevas prestaciones al proyecto, y en poco tiempo el proyecto se realiza bajo una planificación demasiado optimista.
- El proyecto simplemente se estima mal.

3.1.2.- EFECTOS DE LA PLANIFICACIÓN EXCESIVAMENTE OPTIMISTA

Exactitud de la planificación. El uso de la planificación excesivamente optimista reduce la exactitud de la planificación.

Calidad de la planificación del proyecto. Una planificación demasiado optimista disminuye la efectividad de la planificación al suministrar suposiciones erróneas en la fase de planificación, planificación a nivel de personal, selección de la plantilla, planificación de la configuración de la plantilla, planificación del desarrollo de módulos, planificación de la integración, planificación de la prueba y planificación de la documentación.

Las suposiciones erróneas dan lugar a planes de proyectos poco eficientes. Las planificaciones excesivamente optimistas no tienen como resultado planes que se retrasan sólo por un poco. La estimación media de proyectos pequeños se desvía en más de un 100 por 100 [Standish Group, 1994]. La media de un proyecto grande es de un año de retraso [Jones, 1994]. Con errores de esta magnitud, los proyectos en general tienen planificaciones que resultan virtualmente inútiles.

Apego al plan. Todas las organizaciones de software olvidan sus planes y trabajan a su aire cuando se encuentran con problemas en la planificación [Humphrey, 1989]. Establecer una planificación optimista e irrealizable aumenta el riesgo de que un proyecto se ejecute sin un plan.

Reducción del alcance del proyecto. Una planificación excesivamente optimista puede hacer que se dedique poco tiempo a las actividades iniciales de análisis de requisitos y diseño.

Ya es suficientemente difícil saber si realmente se ha hecho un buen trabajo en los requisitos y en el diseño, y una planificación corta incita a no pensar mucho en ello. el efecto normal de una planificación corta es que el proyecto pasa superficialmente por los requisitos y el diseño, no finalizando el trabajo o haciéndolo mal.

Dispersión de actividades. Una planificación demasiado optimista puede desviar la atención de los responsables hacia actividades distintas de las que hacen avanzar el proyecto.

Cada vez que un proyecto cambia la fecha de entrega, normalmente pasa por un ejercicio de replanificación, y eso también consume tiempo de los programadores.

Relaciones con el cliente. Es mejor ser optimista que pesimista; un pesimista será sorprendido agradablemente más veces. En el tema de las relaciones con el cliente, cuando comienza a parecer que un proyecto no cumplirá su fecha de entrega optimista, clientes, responsables y usuarios finales reciben una sorpresa desagradable. De nuevo, la atención del responsable del proyecto se desvía desde la gestión del proyecto hacia la gestión de la relación con el cliente, y la atención del desarrollador se desvía del trabajo real que está realizando, hacia la generación de señales de progreso que tranquilicen al cliente.

A largo plazo, la planificación optimista desgasta la relación con el cliente, porque los clientes pierden confianza en los responsables y desarrolladores excesivamente optimistas.

Convergencia prematura. Cuando se trabaja en un proyecto planificado para una fecha determinada, unas semanas antes de la fecha de entrega se emprende un esfuerzo importante para preparar el producto para entregarlo. Actividades que consumen tiempo y normalmente no se llevan a cabo hasta justo antes de la entrega:

- Desactivar las herramientas de depuración de código.
- Puesta a punto del producto.
- Eliminar o desactivar prestaciones implementadas parcialmente que no pueden ser terminadas a tiempo para la entrega.
- Implementar versiones rápidas de prestaciones que deben finalizarse ineludiblemente a tiempo para la entrega del producto.
- Corregir los defectos de baja prioridad.
- Pulir archivos de ayuda y documentación de usuario.
- Realizar pruebas exhaustivas del producto.

Estas actividades fuerzan al producto a *converger*. Cuando un proyecto intenta forzar a la convergencia demasiado pronto, no logrará converger, y después habrá que repetir todas estas actividades que consumen tiempo.

Hacer actividades dos veces, cuando se podrían hacer sólo una vez, es ineficiente. La convergencia prematura es una pérdida de tiempo.

Quizás el efecto de la convergencia prematura sobre la moral del desarrollador es el más perjudicial para el proyecto. Marcar el ritmo es importante, y la convergencia prematura quema a los desarrolladores demasiado pronto.

Los proyectos mal gestionados descubren por primera vez sus problemas de planificación cuando sus desarrolladores no son capaces de forzar la convergencia.

A continuación se enumeran algunos de los indicios de convergencia más importantes:

- Los desarrolladores no parecen arreglar los defectos sin dividir el sistema; los cambios pequeños llevan más tiempo del que deberían.
- Los desarrolladores tienen largas listas de cambios “triviales”, los cuales saben que tienen que hacer, pero que aún no han hecho.
- El equipo de pruebas encuentra los defectos con más rapidez que la empleada por los programadores para corregirlos.
- La corrección de defectos genera tantos o más defectos nuevos que los corregidos.
- El equipo de documentación tiene problemas porque el software cambia demasiado deprisa para que sea posible documentarlo adecuadamente.
- Las estimaciones del proyecto son ajustadas demasiadas veces a cantidades similares.

Las planificaciones optimistas tienden a hacer múltiples y prematuros intentos de convergencia, los cuales prolongan las planificaciones.

3.2.- Presión excesiva en la planificación

La primera reacción de los responsables y los clientes cuando descubren que no están cumpliendo una planificación optimista es cargar más presión en la planificación sobre los desarrolladores e insistir en que realicen más horas extras. La presión excesiva en la planificación ocurre aproximadamente en un 75 por 100 de los proyectos grandes, y cerca del 100 por 100 en todos los proyectos muy grandes [Jones, 1994]. Casi el 60 por 100 de los desarrolladores informa de un aumento en el nivel de tensión que sienten [Glass, 1994c].

La presión en la planificación ha llegado a ser tan arraigada en el ámbito del desarrollo del software que muchos desarrolladores la han aceptado como un hecho establecido. Algunos desarrolladores ya ni siquiera son conscientes de que la presión extrema en la planificación que experimentan podría ser distinta. La presión en la planificación perjudica, y mucho, a continuación se detallan algunos de los aspectos que se ven perjudicados.

3.2.1.- VÍCTIMAS DE LA PRESIÓN EXCESIVA EN LA PLANIFICACIÓN

Calidad. Se ha determinado que alrededor de un 40 por 100 de todos los errores del software son causa de la tensión; estos errores se podrían haber evitado con una planificación apropiada y no provocando tensión en los desarrolladores [Glass, 1994c]. Cuando la presión en la planificación es extrema, se detectan sobre cuatro veces más errores en el producto respecto a productos desarrollados con una presión menos extrema [Jones, 1994].

Con una presión extrema en la planificación, los desarrolladores también incrementan la presión interna. y son más sutiles para dedicarse a concentrarse en su propio trabajo, descuidando las actividades de garantía de calidad. Por ejemplo, cuando se enfrentan con la opción de elegir entre revisar código de otros o realizar sus propias rutinas.

Los proyectos que aplican una presión excesiva en la planificación y escatiman en la calidad sufren un vuelco cuando descubren que lo que realmente han subestimado es la planificación.

Azar. Como una planificación excesivamente optimista es imposible de alcanzar con los métodos de desarrollo eficiente normales, los directivos y desarrolladores del proyecto sienten la tentación de apostar al azar en vez de correr riesgos calculados.

Los proyectos de software necesitan correr riesgos calculados, pero no riesgos que sólo se puedan gestionar cerrando los ojos y esperando que funcionen. La presión en la planificación contribuye a una gestión deficiente de los riesgos y a errores que ralentizan el desarrollo.

Motivación. A los programadores de software les gusta trabajar. Un poco de presión en la planificación resultante de una planificación ligeramente optimista pero factible puede motivar. Pero en algún punto, la planificación optimista cruza el umbral de la credibilidad, y en ese punto la motivación decae rápidamente.

Una planificación excesivamente optimista hace que los desarrolladores realicen esfuerzos hercúleos que serán considerados como fracasos por no cumplir una planificación imposible, incluso cuando han llevado a cabo una planificación nada extraordinaria. Los programadores saben esto, y a menos que medie juventud e inexperiencia, no se comprometerán para llevar a cabo una planificación fuera de alcance. Alguien que intente motivar forzando la aceptación de una planificación irrealizable, lograra exactamente lo contrario de lo deseado.

Creatividad. Muchos aspectos del desarrollo del software requieren ideas creativas. La creatividad requiere pensar mucho y una gran persistencia cuando la solución deseada no aparece inmediatamente, lo que requiere una motivación interna. La motivación externa excesiva (es decir, estrés) reduce la motivación interna, y a su vez la creatividad [Glass, 1994a].

Además de reducir el estímulo creativo, un entorno de presión es simplemente el peor entorno para tener ideas creativas. La reflexión requerida para una solución progresista necesita un estado mental contemplativo y relajado.

Con un mismo conjunto de requisitos, los programadores crearán soluciones que varían sobre un factor de 10 en las cantidades de código que requieren [DeMarco y Lister, 1989].

Agotamiento. Si se abusa de las horas extras en un proyecto, los desarrolladores se verán afectados en el próximo proyecto. Los desarrolladores se ocuparán de cosas de poca importancia pasados algunos meses del empujón inicial.

Si la planificación presiona demasiado a los desarrolladores, el agotamiento se puede experimentar en el proyecto actual, en vez de en el próximo.

Cambio de personal. Las planificaciones extremadamente optimistas y la presión resultante en la planificación tienden a causar cambio voluntario excesivamente alto de personal, y las personas que dejan el proyecto tienden a ser las más competentes, con las mejores características de rendimiento [Jones, 1991].

Encontrar y formar a sustitutos prolonga la planificación. Hoy en día debido a la demanda el problema es de mayores dimensiones.

Desarrollo rápido a largo plazo. Las horas extras excesivas eliminan el tiempo libre que los desarrolladores invertirían en su desarrollo profesional. Los desarrolladores que no progresan, no aprenden métodos nuevos, y eso perjudica a la capacidad de desarrollo rápido a largo plazo en la organización.

Relación entre desarrolladores y directivos. La presión en la planificación aumenta las diferencias entre desarrolladores y directivos.

Alimenta la tendencia existente entre los desarrolladores de creer que los directivos no los respetan, no se preocupan por ellos, y no saben lo suficiente sobre desarrollo de software como para saber cuándo están pidiendo algo que es imposible. Las malas relaciones llevan a bajar la moral, perder la comunicación y otras situaciones contrarias a la productividad.

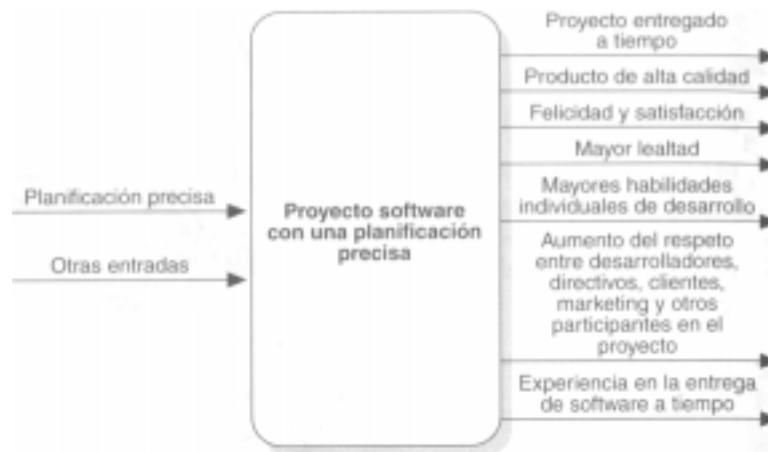
3.2.2.- PUNTOS CRUCIALES

Algunas personas piensan que los proyectos software deberían planificarse de forma optimista, ya que el desarrollo del software debe ser más bien una aventura que un ejercicio monótono de ingeniería. Estas personas alegan que la presión en la planificación ayuda a la emoción. Nada más falso, sin sentido y cercano al fracaso que estas alegaciones.

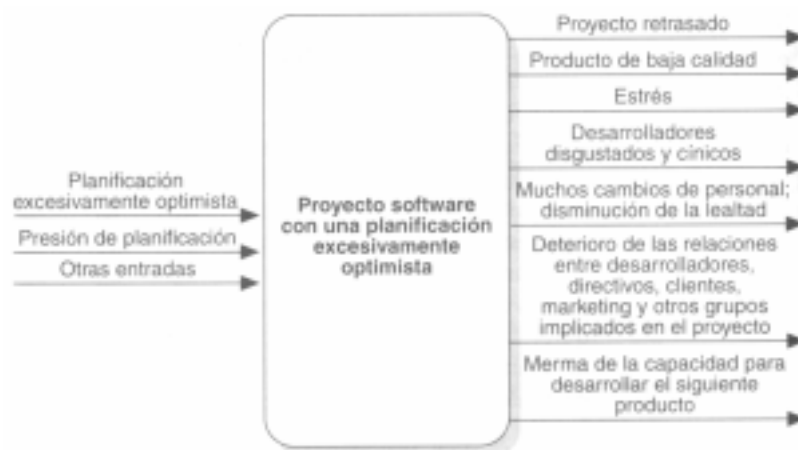
En *Quality Software Management*, Gerald Weinberg sugiere pensar en los proyectos software como en sistemas (Weinberg). Cada sistema tiene unas entradas y genera unas salidas. El diagrama del sistema para un proyecto que ha sido planificado con precisión sería similar al de la figura 3.1.

Desafortunadamente, el diagrama del sistema para un proyecto que ha sido planificado de forma excesivamente optimista se parecerá más al esquema de la figura 3.2.

Cuando se comparan los dos sistemas, se ve de manera clara que uno es notablemente mejor al otro. La mayoría de las personas se sentirán contentas con los resultados de un proyecto planificado con precisión. Y a la mayoría de las personas no le gustarán los resultados obtenidos en el proyecto con una planificación excesivamente optimista.

Figura 3.1. Diagrama del sistema para un proyecto con una planificación precisa.

Fuente: [Mc Connell, 1996]

Figura 3.2. Diagrama del sistema para un proyecto con una planificación excesivamente optimista

Fuente: [Mc Connell, 1996]

El método de planificación excesivamente optimista va contra la planificación efectiva, se inmiscuye en la vida personal del desarrollador, desgasta las relaciones con los clientes, contribuye a aumentar el cambio de personal, contribuye a bajar la calidad y daña a la empresa, impidiendo un progreso profesional y dando la sensación de que los desarrolladores de software no pueden cumplir lo que prometen.

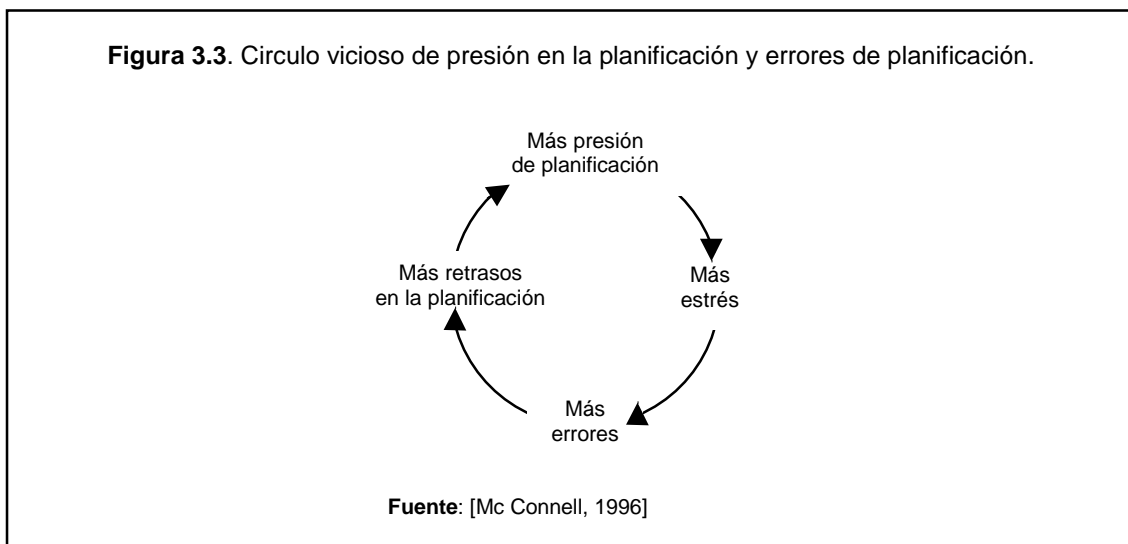
Una planificación excesivamente optimista simplemente no funciona, no produce planificaciones más cortas sino todo lo contrario.

La planificación real más corta es el resultado de la planificación más precisa posible. Un proyecto que tiene problemas de planificación sólo necesita buscar la fuente de sus problemas en una planificación inicial irrealizable.

3.3.- Disminución de la presión de la planificación

La presión en la planificación parece ser endémica en el desarrollo del software, y ha sido perjudicial, pensando a corto plazo, en dos niveles. A nivel local, ha alentado la toma de atajos en proyectos específicos. En un nivel más global, ha contribuido a una mentalidad curiosa sobre la presión en la planificación. La gente ve la presión en la planificación un problema exclusivo del proyecto actual, aunque han sufrido presión en la planificación en todos los proyectos anteriores, y aunque ha sido una de las características definitorias de la industria del software durante al menos 30 años.

Irónicamente, no se puede resolver el problema del desarrollo rápido hasta que se resuelva el problema de la presión en la planificación. La presión en la planificación crea un círculo vicioso (figura 3.3) de más tensión, más errores, más errores de planificación, e incluso de más presión en la planificación.



Hay tres factores que convergen para conformar el núcleo de los problemas asociados con el establecimiento de las planificaciones del software:

- *Ilusiones.* Clientes, responsables y usuarios finales desean, como es de esperar, obtener el máximo partido de su dinero, y desean obtenerlo lo antes posible. La mayoría de las planificaciones de los proyectos software son ambiciosas. La mayoría no son de tipo medio. Anteriormente se han proporcionado las que deberían ser razones necesarias para abandonar las ilusiones sobre las planificaciones del software.
- *Pocos conocimientos sobre la historia de la estimación del software o los efectos reales de la planificación excesivamente optimista.* El software no puede ser bien estimado en sus estados iniciales (lógicamente). Sin embargo, se deja que se fuercen estimaciones poco realistas.
- *Poca habilidad para negociar.* Generalmente los desarrolladores son buenos para estimar, pero malos defendiendo sus estimaciones.

Los desarrolladores tienden a ser malos negociadores por una serie de razones.

En primer lugar, por lo general, los desarrolladores son introvertidos y las relaciones sociales competitivas no suelen ser su fuerte.

En segundo lugar, las planificaciones del software normalmente se establecen mediante negociaciones entre desarrollo y gestión o desarrollo y marketing. El personal de marketing suele ser 10 años mayor y negocia para vivir, es decir, tienden a ser negociadores maduros y profesionales [Weinberg, 1994].

En tercer lugar los desarrolladores tienden a oponerse enérgicamente a los trucos de negociación. Tales trucos ofenden su sentido de calidad y claridad técnica. Los desarrolladores no ofrecerán unas estimaciones iniciales desproporcionadas incluso cuando sepan que los clientes, comerciales o responsables comenzarán con unas posiciones de negociación iniciales para comenzar a regatear.

Los desarrolladores necesitan ser mejores negociadores, y aprender a negociar planificaciones eficientemente.

Negociación conveniente. Un buen punto para comenzar es el método de negociación conveniente descrito en *Getting to Yes* [Fisher y Uri, 1981].

Este método no confía en los trucos de negociación, pero explica cómo responder a los trucos cuando otros los utilizan. Se basa en la idea de crear alternativas satisfactorias para todas las partes.

No intenta derrotar a la persona con la que se está negociando: intenta cooperar de forma que ambos puedan ganar.

Es una estrategia abierta, no importa que las otras partes conozcan el método, de hecho, funciona mejor cuando todas las partes involucradas lo conocen y lo utilizan.

La estrategia de negociación conveniente consta de cuatro partes que tratan sobre personas, intereses, opciones y criterios:

- Separar las personas del problema.
- Centrarse en los intereses, no en las posiciones.
 - Apelar a la velocidad de desarrollo verdadera.
 - Apelar al aumento en la probabilidad de éxito.
 - Recurrir a hechos históricos.
- Inventar opciones para beneficio mutuo.
- Insistir en la utilización de criterios objetivos.
 - No negociar la propia estimación.
 - Insistir en que la estimación sea preparada por alguien cualificado.
 - Insistir en un procedimiento de estimación racional.
 - Resistir a las presiones.

Este método viene comentado más ampliamente en el punto cuarto de anexo I, "Gestión Theory-W" donde se utiliza para fijar una estrategia satisfactoria para todas las partes implicadas en el proyecto, donde todos ganen.

4.- Motivación

La motivación es indudablemente la mayor influencia individual sobre como trabajan las personas. La mayoría de los estudios de productividad han encontrado que tiene mayor influencia en la productividad que cualquier otro factor [Boehm, 1981]. La motivación juega un papel crítico en el desarrollo, pero es un factor intangible: es difícil de cuantificar, y suele considerarse después de otros factores menos importantes, pero más fáciles de medir. Cualquiera sabe que la motivación es importante, pero pocos hacen algo al respecto.

4.1.- Motivaciones típicas del desarrollador

Personas diferentes se motivan con factores diferentes, y los desarrolladores no se motivan siempre con los mismos factores que las personas en general. La siguiente tabla es una lista ordenada de factores motivacionales para desarrolladores, directivos y personas en general. Es un resumen estadístico, cualquier desarrollador podría coincidir realmente con cualquiera de las otras columnas. Son datos antiguos, algunos factores han cambiado debido a variaciones en las condiciones económicas. Pero lo principal es que la tabla recoge ideas importantes sobre las diferencias entre desarrolladores, sus directivos y las personas en general.

Analistas programadores	Directivos de programadores	Personas en general
1. Realización	1. Responsabilidad	1. Realización
2. Posibilidad de superación	2. Realización	2. Reconocimiento
3. El propio trabajo	3. El propio trabajo	3. El propio trabajo
4. Vida personal	4. Reconocimiento	4. Responsabilidad
5. Oportunidad de supervisión técnica	5. Posibilidad de superación	5. Ascenso
6. Ascenso	6. Relaciones interpersonales, subordinados	6. Salario
7. Relaciones interpersonales, igual nivel	7. Relaciones interpersonales, igual nivel	7. Posibilidad de superación
8. Reconocimiento	8. Ascenso	8. Relaciones interpersonales, subordinados
9. Salario	9. Salario	9. Posición social
10. Responsabilidad	10. Relaciones interpersonales, superiores	10. Relaciones interpersonales, superiores
11. Relaciones interpersonales, superiores	11. Políticas de la compañía y de administración	11. Relaciones interpersonales, igual nivel
12. Seguridad en el trabajo	12. Seguridad en el trabajo	12. Oportunidad de supervisión técnica
13. Relaciones interpersonales, subordinados	13. Oportunidad de supervisión técnica	13. Políticas de la compañía y de administración
14. Políticas de la compañía y de administración	14. Posición social	14. Condiciones de trabajo
15. Condiciones de trabajo	15. Vida personal	15. Vida personal
16. Posición social	16. Condiciones de trabajo	16. Seguridad en el trabajo

Fuente: adaptado de [Boehm, 1981]

- *Comparados con las personas en general*, los desarrolladores están mucho más motivados por la posibilidad de superación, vida personal, oportunidad de supervisión técnica y relaciones interpersonales con sus semejantes. Están mucho menos motivados por su posición social, relaciones interpersonales con los subordinados, responsabilidad y reconocimiento.
- *Comparados con sus directivos*, los desarrolladores están un poco más motivados por la posibilidad de superación, vida personal y oportunidad de supervisión técnica. Y mucho menos motivados por la responsabilidad, reconocimiento y relaciones interpersonales con los subordinados.

Las comparaciones entre los desarrolladores y directivos son particularmente interesantes y ayudan a explicar algunos de los motivos de la falta de comunicación que existe entre ellos. Si un directivo intenta motivar a sus desarrolladores como le gustaría que le motivasen a él, fallará. Los desarrolladores se preocupan poco de la responsabilidad o reconocimiento para la motivación. Para motivar a los desarrolladores hay que hacer hincapié en los estímulos técnicos, autonomía, posibilidad de aprender, uso de nuevas técnicas, participación en la planificación y respeto de sus vidas personales.

Otra fuente de conocimiento dentro de las motivaciones del desarrollador proviene del estudio realizado para determinar los tipos de personalidad de los desarrolladores, utilizando el test Myers-Briggs Type Indicator (MBTI), a continuación se explica cómo usarlo para la motivación de los desarrolladores, el anexo III amplía la información del test. El MBTI mide las preferencias de las personas en cuatro dimensiones y propone una clasificación con cuatro letras. Las categorías son:

- Extraversión (E) o introversión (I).
- Sentido (S) o intuición (N).
- Pensamiento (P) o sentimiento (F).
- Opinión (J) o percepción (P).

Hay 16 combinaciones de 4 letras, o sea, 16 tipos de personalidad.

Sin demasiada sorpresa, dos estudios [Lyons, 1985; Thomsett, 1990] han encontrado que los profesionales del software son mucho más “introvertidos” que las personas en general. Introvertidos en el contexto del MBTI significa que la persona está más interesada en el mundo interior de las ideas que en el mundo externo de las personas y cosas.

Esta tendencia hacia una orientación interior parece coincidir con los datos de la tabla de la página anterior, que muestra que los desarrolladores en general están más interesados en la posibilidad de superación que los demás grupos y menos en la posición social y el reconocimiento.

Los mismos estudios descubrieron que el 80 por 100 de los profesionales del software prefieren el pensamiento (T) sobre el sentimiento (F). Los Ts prefieren tomar decisiones basadas en unas bases lógicas e impersonales más que en valores personales subjetivos. Esta inclinación lógica y planificada se refuerza por la preferencia de los profesionales del software en la opinión (J) sobre la percepción (P). Los Js tienden a vivir de una forma ordenada y planificada, mientras que los Ps tienden a ser más flexibles y adaptables.

La implicación de esta preferencia es clara: con un desarrollador lo mejor es usar argumentos lógicos. Por ejemplo, establecer objetivos aparentemente imposibles podría funcionar para los Fs, quienes podrían encontrar tales objetivos inspiradores, pero los Ts los rechazarán por ilógicos. Por ello es raro que el grupo de desarrolladores responda positivamente a un objetivo de planificación imposible.

Las diferentes formas de motivación, funcionan con personas diferentes. Las generalidades sobre motivación pueden proporcionar ideas generales, pero se tendrá más éxito si se intenta identificar cual es la motivación más efectiva para cada uno individualmente. Hay que intentar ponerse en el lugar de cada uno de los miembros del equipo y comprender como piensa (empatía).

4.2.- Uso de los cinco factores de motivación

4.2.1.- REALIZACIÓN

A los desarrolladores de software les gusta trabajar, la mejor forma de motivar a los desarrolladores es proporcionarles un entorno que les facilite centrarse.

Propiedad. La propiedad es una clave para la motivación por realización. Las personas trabajarán más duro para alcanzar sus propios objetivos que los de las demás personas.

Definición de objetivos. La definición de objetivos es otra clave para la motivación por realización. Establecer los objetivos explícitamente es un paso obvio y simple para un buen y rápido desarrollo, y es fácil de supervisar.

Los desarrolladores no pueden responder a los objetivos que cambian diariamente o que son imposibles de cumplir en conjunto. Los desarrolladores hacen lo que se les pida que hagan. Tienen una gran motivación por realización, trabajan en los objetivos que se les especifican, pero hay que especificar esos objetivos y ser conciso. Si a un equipo se le dan varios objetivos a la vez, no es probable que los hagan todos bien. Un estudio de ITT demostró que la productividad disminuía claramente cuando había muchos objetivos [Vosburgh *et al.*, 1984].

Definir demasiados objetivos al mismo tiempo es un problema común. Para obtener mejores resultados, hay que seleccionar un objetivo y dejar claro que es el más importante.

4.2.2.- POSIBILIDAD DE SUPERACIÓN

Uno de los aspectos más excitantes de ser un desarrollador de software es trabajar en un campo que está cambiando constantemente.

Hay que considerar la naturaleza de la industria elegida por los desarrolladores para trabajar en ella, y no sorprenderse de la motivación por posibilidad de superación.

Una organización puede influir en esta motivación dando a los desarrolladores las oportunidades de crecer en sus proyectos. Esto requiere alinear los objetivos de crecimiento de la organización con los objetivos de superación individuales. Barry Boehm [1981] lo indica de esta forma:

El principio de la progresión en la carrera indica que está en una organización cuyo principal interés es ayudar a las personas a determinar cómo desean prosperar profesionalmente, y proporcionarles oportunidades para desarrollarse en esas direcciones. Esto puede parecer un principio obvio pero en la práctica hay muchas organizaciones de software que siguen principios fuertemente opuestos.

Si se elimina la posibilidad de superación al equipo, se elimina al mismo tiempo la motivación del equipo.

Una empresa puede mostrar interés en el progreso profesional de sus programadores de varias (no están todas) formas:

- Cursos de desarrollo profesional.
- Tiempo extra para asistir a clases o estudiar.
- Dinero para la compra de libros profesionales.
- Asignando a los desarrolladores a proyectos que aumentarán su experiencia.
- Asignando un tutor a cada desarrollador.
- Evitando presiones excesivas en la planificación.

Las compañías que están en el grupo de cabeza de sus sectores en calidad y productividad, normalmente dedican 2 semanas al año para la preparación de los desarrolladores del software y tres semanas para los responsables del software [Jones, 1994].

Centrarse en la superación personal puede tener consecuencias a corto y largo plazo sobre la productividad de la organización. A corto plazo, aumentará la motivación del equipo, haciendo que trabajen más duro. A largo plazo, aumentará la posibilidad de que la organización atraiga y mantenga a personas con grandes aptitudes.

4.2.3.- EL TRABAJO EN SÍ

Richard Hackman y Greg Oldham mantienen que, generalmente, la motivación interna de las personas parte de tres fuentes: su trabajo debe tener sentido; debe tener responsabilidad por el resultado del trabajo; y debe conocer los resultados reales de las actividades de su trabajo [Hackman y Oldham, 1980].

Hackman y Oldham identificaron cinco dimensiones del trabajo en sí que contribuyen a esas fuentes de motivación. Las tres primeras características de este trabajo contribuyen a ver el significado que las personas encuentran en su trabajo:

- *La diversidad de técnicas* es el grado en el que el trabajo requiere que se ejerciten una serie de técnicas, de forma que se pueda evitar el aburrimiento y la fatiga. Aún en trabajos no muy significativos ni importantes, la variedad encuentra un significado al trabajo.
- *La identidad de la tarea* es el grado en el que el trabajo requiere que se cubra una parte completa. Las personas ponen más interés cuando tienen que hacer un trabajo completo y sienten que su colaboración es importante.
- *La importancia de las tareas* es el grado en el que el trabajo afecta a otras personas y contribuye al bienestar social. Las personas necesitan sentir que el producto final tiene un valor.

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

- *Autonomía* es el grado en el que se tiene el control sobre los medios y métodos que se utilizan para realizar el trabajo. Cuanto mayor sea la autonomía, mayor será el sentimiento de responsabilidad personal por el resultado del trabajo.

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

- *Realimentación del trabajo* es el grado en que la realización del trabajo en sí proporciona información clara y directa sobre cómo es de efectivo.

Una clave para la motivación es controlar estas cinco dimensiones para crear un trabajo significativo, y posteriormente asignar este trabajo a personas con un gran deseo de realizarlo. El 60 por 100 de la motivación de un desarrollador proviene de la correspondencia entre el trabajo y el desarrollador [Zawacki, 1993].

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

Conseguir algo inusual ha tardado desde semanas a meses. Para un desarrollador de software que no desea otra cosa que desarrollar software, puede ser increíblemente frustrante y desmotivador tener que perder el tiempo con tareas burocráticas o peleándose para conseguir material normal de oficina.

4.2.4.- VIDA PERSONAL

El impacto motivacional de la vida personal de un desarrollador es probablemente el factor motivacional más duro de comprender por un directivo. El resultado es que los directivos premian en ocasiones a sus mejores desarrolladores asignándoles los proyectos más complejos y de mayor presión. Para los directivos, la responsabilidad extra, puede ser un placer, y la disminución de la vida personal no importaría mucho. Para un desarrollador, la responsabilidad extra es más un engaño que un placer, y la disminución de la vida personal es una fuerte pérdida. El desarrollador interpreta el “premio” del directivo como un castigo.

No se puede hacer mucho para utilizar las vidas personales como factores de motivación, excepto planificar proyectos de manera realista, para que los desarrolladores tengan tiempo para su vida personal, respetando periodos vacacionales, y siendo susceptible a peticiones ocasionales de tiempo libre durante el trabajo.

4.2.5.- OPORTUNIDAD DE SUPERVISIÓN TÉCNICA

Para un desarrollador una oportunidad de revisión técnica representa una realización. Una oportunidad de supervisión técnica implica que el desarrollador ha alcanzado un nivel de experiencia técnica suficiente para dirigir a otros.

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

- En un proyecto, asignar a cada persona como responsable técnico de un área del producto en particular.
- Asignar a cada persona la responsabilidad técnica de un área del proceso en particular.
- Hacer que casi todos los desarrolladores sean monitores.

4.3.- Uso de otros factores de motivación

4.3.1.- PREMIOS E INCENTIVOS

Los desarrolladores se cansan de trabajar para compañías desagradecidas, y por tanto, los premios son una importante motivación a largo plazo. Pero los premios en metálico tienen que darse con cuidado. Los desarrolladores son buenos en matemáticas, y pueden calcular cuando un premio no es proporcional al sacrificio que han realizado.

Los malos sistemas de recompensa son sistemas en los que la organización aumenta un 6 por 100 a sus mejores trabajadores y un 5 por 100 a los mediocres. Normalmente, los mejores trabajadores se sienten frustrados y se van [Boehm, 1987a].

También es importante dar un premio simplemente como un gesto de reconocimiento más que como un incentivo. Al menos dos docenas de estudios realizados durante los últimos treinta años han demostrado concluyentemente que las personas que esperan recibir un premio por realizar bien su trabajo no lo hacen tan bien como aquellos que no esperan recibir nada [Kohn, 1993]. El trabajo en sí es el motivador más importante y cuanto más énfasis se le da a una recompensa externa, menos interés tiene el desarrollador en el trabajo en sí y se pierde más motivación potencial.

A continuación se enumeran algunos posibles gestos de apreciación que suelen ser apropiados:

- Una felicitación sincera dirigida a un logro específico.
- Productos de mercadotecnia relativos al equipo.
- Placas, certificados, etc. en broma o serios y cosas por el estilo.
- Acontecimientos especiales para celebrar logros significativos.
- Excepciones en la política de la compañía para el equipo.
- Cursos especiales (fuera del lugar de trabajo).
- Patrocinio de conferencias.
- Ascenso de categoría.
- Gratificaciones especiales.

Peters y Waterman [1982] informan que las compañías que han logrado estar en la primera mitad de sus campos durante más de 20 años, han hecho un uso extensivo de incentivos no monetarios.

Como con cualquier muestra de apreciación, la intención es lo que cuenta. Hay que asegurarse de que los premios indican “apreciación” más que “incentivos” o “manipulación”.

4.3.2.- PROYECTOS PILOTO

En uno de los más famosos experimentos sobre motivación y productividad, Elton Mayo y sus socios pasaron una serie de pruebas sobre la productividad de los trabajadores de Hawthorne Works of the Western Electric Company en Chicago desde 1927 hasta 1932. Su objetivo era determinar el efecto de la iluminación en la productividad. Primero aumentaron la luz y la productividad subió. Luego bajaron las luces, y la productividad subió de nuevo. Luego mantuvieron las luces constantes, y la productividad subió nuevamente [Boehm, 1981].

Después de muchos más experimentos, Mayo llegó a la conclusión de que no tenía nada que hacer con los niveles de luz. Concluyó que el simple hecho de llevar a cabo los experimentos había incrementado la productividad.

Para un responsable o un directivo el “efecto Hawthorne” es oro puro, no importa si la mejora en la productividad procede de la nueva tecnología o del efecto Hawthorne. Si los resultados experimentales están desviados, serán desviados hacia la parte positiva, que es lo que se pretendía de cualquier modo.

La implicación para los proyectos software es clara. Hay que ejecutar todo proyecto software como un experimento, como un proyecto piloto. Hay que intentar aplicar alguna metodología o tecnología nuevas en cada proyecto nuevo y asegurarse de que el equipo tiene conocimiento de que es un proyecto piloto.

Al igual que con los premios, es bueno recordar que hay una línea muy fina entre la motivación y la manipulación.

4.3.3.- REVISIONES DEL RENDIMIENTO

Una ejecución apropiada de una revisión del rendimiento aumenta de forma significativa la motivación, y la ejecución inapropiada de la revisión del rendimiento reduce de forma significativa la motivación.

El presidente de Intel piensa que la revisión del rendimiento:

Es la forma más importante de realimentación de las tareas relevantes que como supervisores podemos proporcionar.

La revisión influirá en el rendimiento de un subordinado (positiva o negativamente) durante mucho tiempo, lo que hace de la estimación una de las actividades de mayor influencia de los directivos

[Grove, 1982]

Pero, hay que asegurarse de que las revisiones en las que se está metido aumentan más que reducen la motivación.

4.4.- Destructores de la moral

Tan importantes como los factores de motivación son los factores que desmotivan. A continuación se explican algunos de los más importantes y de mayor incidencia negativa en la motivación.

4.4.1.- FACTORES DE HIGIENE

Los factores de higiene son las condiciones básicas que un trabajador necesita para trabajar efectivamente. En el mejor caso, los factores de higiene no crean descontento. En el peor caso, su ausencia crea descontento.

Los buenos desarrolladores tienden hacia organizaciones que proporcionan un entorno de trabajo en el que se puede ser productivo.

A continuación se exponen una serie de factores de higiene para los desarrolladores de software:

- Iluminación, calefacción y aire acondicionado apropiados.
- Espacio adecuado para la mesa de despacho y la estantería.
- Tranquilidad suficiente para poder concentrarse.
- Intimidación suficiente para evitar interrupciones no deseadas.
- Acceso a equipamiento de oficina.
- Acceso rápido a los suministros de oficina.
- Acceso no restringido al ordenador.
- Equipo informático actualizado.
- Reparación inmediata o casi inmediata de los equipos informáticos averiados.
- Soporte actualizado de comunicaciones.
- Herramientas de software pertinentes.
- Periféricos adecuados.
- Manuales de referencia y variedad de publicaciones.
- Libros de consulta auxiliares y ayudas de referencia en línea.
- Formación mínima.
- Copias legales del software utilizado.
- Libertad al establecer el horario de trabajo, tanto general como específicamente.

4.4.2.- OTROS DESTRUCTORES DE LA MORAL

Manipulación de la directiva. Los desarrolladores son susceptibles a ser manipulados por los directivos. Tienden a enfrentarse a temas importantes, y desean que la directiva trate a los desarrolladores de forma sincera.

Cualquier directivo podría tener buenas razones para pedir una fecha límite o una petición similar, sin tener que dar explicaciones. Los directivos de un nivel inferior podrían no entender las razones por las que se ha establecido la fecha límite. Razones que podrían parecer ser evasivas y manipuladoras (y los desarrolladores responden mal a esto).

Salvo circunstancias extremas cualquier directivo que pida a los desarrolladores que realicen trabajo extra en un proyecto les debe dar una clara explicación.

Los directivos que se esfuerzan por no proporcionar detalles sobre porqué una fecha límite es importante, deberían recordar que sus explicaciones probablemente resulten desalentadoras para los desarrolladores.

Presión excesiva de la planificación. Como se vio en el capítulo 3, si la fecha límite es real, podría no ser realista. Una de las formas más rápidas de bajar la motivación a cero es presentar a los desarrolladores una fecha límite imposible. Pocas personas trabajarán duro para alcanzar una fecha límite que saben que es imposible, especialmente si son del tipo T MBPI (aquellos que responden más a la lógica que a las emociones).

Falta de apreciación de los esfuerzos de desarrollo. En ocasiones los directivos no ven la cantidad de trabajo que los desarrolladores están realizando, de forma que piensan que no están haciendo mucho y, por tanto, deben obligarlos. En realidad, los desarrolladores están extremadamente automotivados, trabajan duro y muchas horas, y encontrarse acusados de “holgazanear” es muy desalentador. Si se desea que los desarrolladores hagan más de lo que realmente ponen de manifiesto en la oficina, nunca jamás hay que decirles que no están trabajando duro cuando si lo están haciendo.

Participación de directivos sin preparación técnica. Los desarrolladores pueden ser motivados por directivos técnicamente incapaces, siempre que esos directivos reconozcan que no tienen preparación técnica y limiten el control del proyecto a las decisiones de índole no técnica. Intentar meterse en decisiones técnicas que no se comprende, significa convertirse en el blanco de las bromas del equipo de desarrollo, y la mayoría de las personas no pueden ser motivadas por alguien a quien no respetan.

No involucrar a los desarrolladores en decisiones que les afectan. No involucrar a los desarrolladores en decisiones que les afectan da la impresión de que el responsable no presta atención al equipo de desarrollo, o que ese directivo no lo respeta lo suficiente como para desear su participación. A continuación se muestran algunos ejemplos clásicos en donde la directiva debe comprometer a los desarrolladores si desea mantener alta su motivación:

- Compromiso con las nuevas planificaciones.
- Compromiso con las nuevas prestaciones o con modificaciones de las mismas.
- Contratar a nuevos miembros del equipo.
- Designar a desarrolladores para tareas a corto plazo dentro del proyecto.
- Diseñar el producto.
- Tomar decisiones de equilibrio técnico.
- Cambiar la oficina.
- Cambio de periféricos.
- Cambiar las herramientas de programación.
- Comprometerse a entregar productos que podrían o no haber sido planificados por el equipo.
- Comprometerse a nuevos proyectos de desarrollo.

A un directivo, podría parecerle necesario realizar el compromiso o el cambio descrito en cada uno de los casos anteriores. Un directivo tiene derecho a hacer eso. También tiene derecho a convertir en nula la motivación de su equipo. Si se desea que el equipo tenga una motivación menor de cero, hay que intentar solicitar las ideas del equipo para encubrir una decisión, después de haberla tomado. Esa combinación de fallos y de manipulación para comprometer a los desarrolladores es especialmente desmotivante. Si se desea mantener alta la motivación, hay que involucrar al equipo de desarrollo en la decisión antes de comprometerse o de aprobar el cambio.

Barreras de la productividad. Si el entorno se establece de forma que se frustra el esfuerzo del mejor desarrollador para ser productivo, se puede estar seguro de dañar su motivación. Hay que intentar eliminar las barreras de la productividad, de forma que el equipo pueda centrarse en el trabajo en vez de en salvar las distracciones.

Baja calidad. Los desarrolladores derivan parte de su amor propio a partir de los productos que desarrollan. Si desarrollan un producto de gran calidad se sienten bien y al revés. Para que un desarrollador se motive por orgullo de propiedad, hay que ser propietario del trabajo, y hay que estar orgulloso del mismo. Pocos desarrolladores pueden sentir orgullo como respuesta al reto de desarrollar un producto de baja calidad en el menor tiempo posible. Porque la mayoría de los desarrolladores están más motivados por la calidad que por la producción total.

Si un directivo insiste en que los desarrolladores disminuyan la calidad para tener una planificación más corta está reduciendo a la mitad la motivación. Con un producto de baja calidad algunos desarrolladores se sentirán realmente mal, aunque entreguen en las fechas límite y reciban bonificaciones.

Si no parece posible que los desarrolladores construyan un producto de alta calidad en el tiempo disponible, hay que dejar que los desarrolladores saquen sus propias conclusiones. Podrían escoger entre diseñar un producto menos absorbente, o desarrollar un producto de menor calidad en el tiempo establecido. A menos que ellos decidan, no se conseguirá nada nuevo intentando forzarlos a desarrollar un producto de baja calidad. Eso no es motivación, y no funciona.

Campañas cargantes de motivación. Con lemas, arengas y otros métodos de motivación es más fácil insultar la inteligencia del desarrollador que motivarlo. Con los desarrolladores de software, funciona mejor un ligero toque.

5.- El equipo de trabajo

5.1.- Usos de los equipos de trabajo en el software

Un equipo de trabajo es algo más que un conjunto de personas que desean trabajar juntas para formar un equipo.

Un equipo es un número de personas pequeño, con habilidades complementarias, que están comprometidas en un propósito, objetivos de rendimiento y con un enfoque comunes, en el que todos sean responsables ante todos.

[Katzenbach y Smith, 1993]

El equipo de trabajo en los proyectos software puede entrar en juego en cualquier número de tareas específicas:

- Desarrollo y revisión de requisitos del proyecto.
- Desarrollo de la arquitectura y las directrices del diseño del proyecto.
- Definición de aspectos del entorno técnico.
- Desarrollo de los estándares de codificación del proyecto.
- Coordinar el trabajo en las partes relacionadas del proyecto.
- Diseño de las partes complicadas del sistema.
- Revisión del diseño y código de los desarrolladores individuales.
- Depuración de las partes más complicadas.
- Prueba de requisitos, diseño y código.
- Auditoría del avance del proyecto.
- Mantenimiento del software, una vez construido.

Aunque algunas de las tareas anteriores las podría realizar una sola persona, se pueden beneficiar de la participación de dos o más mentes, lo cual requiere la interacción entre los miembros del proyecto. Si las mentes trabajan juntas, a veces el total puede ser mayor que la suma de las partes. Si hay desacuerdo, puede ser menor. Un “equipo” existe siempre que dos cabezas juntas funcionan mejor que cada una por separado.

5.2.- Importancia del equipo de trabajo

Los proyectos pequeños pueden pasar sin estudiar cuestiones sobre equipos de trabajo, pero se beneficiarán de su tratamiento. Los proyectos grandes son esfuerzo de grupos, y las características de los grupos juegan un papel muy importante en el éxito.

Variaciones en la productividad del equipo. Los investigadores han encontrado diferencias del orden de 10 a 1 en la productividad individual. Los investigadores también han encontrado diferencias importantes en los niveles de productividad de los equipos. Después de analizar 69 proyectos, Barry Boehm llegó a la conclusión de que los mejores equipos eran al menos 4 veces más productivos que los peores [Boehm, 1981]. De Marco y Lister identificaron diferencias de productividad de 5, 6 a 1 en un estudio de 166 programadores de 18 empresas [De Marco y Lister, 1985].

La diferencia en la productividad se mantenía incluso entre grupos de desarrolladores con niveles de experiencia similares en un factor de 3 a 1 y 4 a 1. Valett y Mc Garry informaron de diferencias de 2 a 1 y 3 a 1 en la productividad entre diferentes proyectos del Laboratorio de Ingeniería de Software de la NASA [Valett y Mc Garry, 1989].

La conclusión es que entre grupos con diferentes conocimientos y diferentes niveles de experiencia, hay una diferencia de 5 a 1 en la productividad. Entre grupos con similares conocimientos y niveles de experiencia, hay una diferencia en la productividad de 2'5 a 1.

Cohesión y rendimiento. Los miembros de los grupos unidos trabajan duro, se divierten con su trabajo, y emplean gran parte de su tiempo centrándose en los objetivos del proyecto.

Los participantes en proyectos con malas dinámicas de equipo normalmente no están centrados y están desmoralizados, y emplean gran parte de su tiempo trabajando en objetivos opuestos.

Lakhanpal [1993] descubrió que la unión del grupo contribuía más a la productividad que las capacidades o experiencia individuales de los miembros del proyecto. Lakhanpal apuntó que los directores normalmente escogen a los miembros del proyecto basándose en el nivel de experiencia y en las posibilidades individuales. Lakhanpal sugiere que los directivos deberían designar a los desarrolladores basándose en sus posibilidades para contribuir primero a formar un equipo unido, y solo luego basarse en sus posibilidades individuales.

5.3.- Creación de un equipo de alto rendimiento

Los equipos productivos a veces se caracterizan por ser equipos que se han ido formando o equipos con una gran unión. Un equipo con una gran cohesión, un alto rendimiento y que se ha ido moldeando a sí mismo, tiene una serie de características comunes:

- Una alta visión u objetivo compartidos.
- Un sentido de identidad de equipo.
- Una estructura dirigida a los resultados.
- Miembros del equipo competentes.
- Un compromiso con el equipo.
- Confianza mutua.
- Interdependencia entre miembros del equipo.
- Comunicación efectiva.
- Un sentido de autonomía.
- Un sentido de enriquecimiento.
- Tamaño del equipo pequeño.
- Un alto nivel de disfrute.

En 1989, Larson y La Fasto publicaron un estudio en el que encontraron una consistencia poco usual entre los atributos de los equipos altamente efectivos [Larson y La Fasto, 1989].

5.3.1.- CARACTERÍSTICAS DE UN EQUIPO DE ALTO RENDIMIENTO

VISIÓN U OBJETIVO COMPARTIDOS

Antes de que el proyecto realmente se ponga en marcha, el equipo necesita "obtener" una visión y unos objetivos comunes.

Si no se tiene una visión compartida, no hay lugar para un equipo de trabajo de alto rendimiento. Larson y La Fasto encontraron que en todos los equipos que funcionaban de forma efectiva los miembros tenían una visión muy clara de cuáles eran los objetivos.

Cuando se está de acuerdo con la visión del proyecto, se simplifica la toma de decisiones. Una visión común crea confianza entre los miembros del equipo, porque saben que todos trabajan para llegar al mismo objetivo. También ayuda a que el equipo se mantenga concentrado y a evitar que se pierda tiempo debido a errores. Un equipo efectivo crea un nivel de confianza y cooperación que le permite alcanzar mayor rendimiento que un conjunto de personas con igual destreza.

Ocasionalmente, un equipo muy unido se encerrará compartiendo una idea que esté de acuerdo con los objetivos de la organización. En este caso, el equipo podría realizar bastante trabajo, pero no del tipo que la organización necesita. Para que un grupo unido sea productivo, necesita tener un enfoque compatible con la organización a la que pertenece.

Trabajo planteado como un estímulo. La visión compartida puede ser algo importante o puede ser relativamente trivial. La visión puede ser virtualmente arbitraria, pero mientras el equipo completo la comparte, servirá al mismo objetivo de ayudar a unir al equipo.

La visión también necesita ser elevada para que surta un efecto de motivación. El equipo necesita que se le presente un desafío. Los equipos de alto rendimiento no se forman alrededor de objetivos difusos.

La respuesta al reto es una reacción emocional, y está muy influenciada por la forma en que se asigne o se describa el trabajo.

Un equipo real necesita realizar una misión, y la presentación del proyecto tiene mucho que ver con que el equipo lo conciba como misión.

SENTIDO DE IDENTIDAD DEL EQUIPO

A medida que los miembros del equipo trabajan juntos hacia su visión común, comienzan a percibir una sensación de identidad del equipo. Algunos equipos tienen una abreviatura, adoptan un código de vestuario del equipo, ... Buscan características comunes que los diferencien de la gente del "montón". Las compañías inteligentes refuerzan el sentido de identidad del equipo.

Los miembros del equipo permiten su sentido de identidad del equipo para eclipsar su identidad individual. Su satisfacción proviene de los logros del equipo. Los objetivos comunes se consideran más importantes que los personales. Tienen la oportunidad de conseguir con el equipo algo que individualmente no podrían.

Los equipos de alto rendimiento suelen desarrollar un sentido de elite junto con el sentido de identidad.

ESTRUCTURA DIRIGIDA POR RESULTADOS

Los equipos se pueden estructurar de forma que se obtenga una salida óptima, o se pueden estructurar de forma que sea casi imposible que produzcan algo. Para un rápido desarrollo, se necesita estructurar el equipo teniendo en cuenta la máxima velocidad de desarrollo.

Las cuatro características fundamentales de una estructura de equipo dirigida por resultados son:

- *Las tareas deben ser claras, y todos deben ser responsables de su trabajo en todo momento.* Después de que se hayan tomado las decisiones, la responsabilidad es fundamental para tomar decisiones efectivas y para llevar a cabo una ejecución rápida.
- *El equipo debe tener un sistema de comunicación efectivo,* que admita el flujo libre de información entre los miembros del equipo. La comunicación debe fluir abiertamente desde y hacia la directiva del equipo.
- *El equipo debe tener algún control sobre el rendimiento individual y la realimentación.* El equipo debe saber a quién recompensar, quién necesita desarrollarse individualmente y quién puede asumir más responsabilidad en el futuro.
- *Las decisiones se deben tomar, si es posible, basándose en hechos más que en opiniones subjetivas.* El equipo necesita asegurarse de que no está influenciado a la hora de interpretar los hechos para no dificultar su funcionamiento.

MIEMBROS COMPETENTES DEL EQUIPO

Al igual que las estructuras del equipo se escogen por razones equivocadas, los miembros del equipo también se suelen escoger por razones equivocadas: por ejemplo, se suelen elegir porque tengan un cierto interés en el proyecto, porque cobran poco o, lo más normal, porque simplemente están disponibles. A la hora de seleccionarlos no se piensa en un buen y rápido desarrollo.

COMPROMISO CON EL EQUIPO

En un equipo efectivo, sus miembros se comprometen con el equipo. Hacen sacrificios personales por el equipo que no harían para las grandes organizaciones. En algunos casos, el requerimiento mínimo para que el equipo tenga éxito es que sus miembros contribuyan con su tiempo y su energía (su esfuerzo), y esto exige un compromiso. Cuando el equipo se compromete, tiene que haber algo por lo que comprometerse. No se pueden comprometer sin tener unos objetivos. La visión, el reto y la identidad del equipo proporcionan lo necesario para que los miembros del equipo se comprometan.

CONFIANZA MUTUA

Larson y La Fasto comprobaron que la confianza consta de 4 componentes:

- Honestidad
- Franqueza.
- Firmeza.
- Respeto.

Si se infringen algunos de estos elementos, aunque sólo sea una vez, la confianza se rompe.

La confianza es menos una causa que un efecto de un equipo efectivo. No se puede forzar a que los miembros de un equipo confíen en los demás. Pero una vez que los miembros del proyecto se comprometen a una visión común, y comienzan a identificarse con el equipo, aprenden a ser responsables y a tener otra responsabilidad. La confianza surgirá si ve que los otros miembros tienen verdadero interés por el grupo.

INTERDEPENDENCIA ENTRE MIEMBROS

Los miembros del equipo confían en la fuerza individual de los demás, y hacen todo lo que sea mejor para el equipo. Todos sienten que tienen posibilidad de contribuir y que su contribución es importante. Todos participan en las decisiones. En resumen, los miembros del equipo llegan a ser interdependientes.

COMUNICACIÓN EFECTIVA

Los miembros de los equipos unidos están al corriente constantemente de lo que les sucede a los demás. Tienen cuidado de comprobar que todos les entienden cuando hablan, y su comunicación se beneficia del hecho de que muestran una visión común y un sentido de identidad.

Los miembros del equipo expresan sus verdaderos “sentimientos”, incluso cuando no se encuentran muy bien. En un entorno que se caracteriza por la confianza y la interdependencia, los miembros del equipo pueden empezar a tratar los problemas cuando tengan la primera noticia, cuando aún quede tiempo para corregirlos de forma efectiva.

SENSACIÓN DE AUTONOMÍA

Los equipos efectivos tienen la sensación de que son libres para poder realizar todo lo que crean necesario, para que el proyecto tenga éxito.

Una de las razones por las que los proyectos funcionan también como lo hacen, es porque se les da a los miembros la oportunidad de hacer lo que vean conveniente, sin preocuparse de que lo que hacen parezca que sea lo correcto. Pueden trabajar sin interferencias. El equipo podría cometer algunos errores, pero las ventajas en motivación compensarán los errores.

Esta sensación de autonomía estará relacionada con el nivel de confianza que sientan por parte de sus directivos hacia ellos. Es indispensable que los directivos confíen en el equipo. Cualquier directivo admitirá a un equipo cuando esté claramente en lo cierto; pero eso no es confianza. Cuando un directivo apoya a un equipo que parece estar equivocado eso es confianza.

SENTIDO DE REFUERZO

Un equipo efectivo necesita sentirse reforzado para llevar a cabo las acciones necesarias para obtener el éxito. La organización no se limita a permitirles hacer todo lo que piensan que es correcto, sino que les apoya a la hora de llevarlo a cabo.

TAMAÑO REDUCIDO DEL EQUIPO

Algunos expertos dicen que para que un equipo cuaje debe tener de 8 a 10 personas [Emery y Emery, 1975; Bayer y Highsmith, 1994]. Si se puede mantener el grupo con ese tamaño, lo mejor es mantenerlo. Si el proyecto necesita tener un grupo con más de 10 miembros, hay que intentar dividirlos en varios equipos, y que cada uno tenga 10 o menos miembros.

El límite de 10 personas se aplica principalmente a los equipos de un solo proyecto. Si se puede mantener un equipo unido a lo largo de varios proyectos, se puede aumentar el tamaño del equipo, siempre que sus miembros compartan una cultura profundamente arraigada.

Por otro lado, es posible que un grupo sea demasiado pequeño para formar un equipo. Emery y Emery apuntaron que con menos de cuatro miembros es difícil tener una identidad de grupo, y el grupo estará bien dominado por las relaciones interpersonales más que por un sentido de responsabilidad de grupo [Emery y Emery, 1975].

ALTO NIVEL DE DISFRUTE

No todo equipo que se divierte es productivo, pero la mayoría de los equipos productivos son divertidos. En primer lugar, a los desarrolladores les gusta ser productivos. Si su equipo admite su deseo de ser productivo, ellos disfrutan con eso. En segundo lugar, las personas emplean por naturaleza más tiempo haciendo cosas que les divierten que cosas que no les divierten, y si emplean más tiempo en eso, realizarán más cosas. En tercer lugar, parte de lo que hace unirse a un equipo es adoptar un sentido del humor propio.

5.3.2.- DIRECCIÓN DE UN EQUIPO DE ALTO RENDIMIENTO

Un equipo compenetrado se puede considerar como un “nosotros”, donde los directivos se encuentran en una posición delicada porque no se pueden integrar completamente en ese “nosotros”, ni pueden considerar al resto del equipo con “ellos”.

A continuación se muestran algunas de las claves para tener éxito en la gestión de un equipo unido:

- Establecer una visión. La visión es lo más importante, y es función de la dirección y el responsable del equipo ponerla en juego.
- Crear un cambio. La dirección reconoce que hay una gran diferencia entre cómo son las cosas en la realidad y cómo deberían ser. Percibe que la visión necesita un cambio, y hace que se produzca dicho cambio.
- Dirigir al equipo como un equipo. Hacer que el equipo se responsabilice de las acciones, en vez de hacer a cada uno responsable de sus acciones individuales. Los miembros del equipo suelen ser más exigentes consigo mismos de lo que les piden sus responsables [Larson y La Fasto, 1989]
- Delegar las tareas en el equipo, de forma clara, estimulante y con ayuda. Se liberan la energía y el talento de los miembros del equipo.
- Dejar al equipo los detalles de cómo realizar las tareas, incluyendo posiblemente la asignación de responsabilidades de trabajo individual.

- Cuando un equipo no funciona bien, piense en el modelo MOI, donde se decide que la mayoría de los problemas que surgen en el equipo se deben a la Motivación, Organización e Información. Hay que intentar eliminar los obstáculos relacionados con estos tres factores.

5.4.- Causas de fallo de los equipos

Falta de visión común. Es difícil que los equipos se formen si no comparten una visión común. Las organizaciones impiden a veces la cohesión de los equipos echando abajo sus ideas. Cuando la idea se viene abajo, el equipo también.

Falta de identidad. Los miembros del equipo podrían estar dispuestos, pero ninguno desempeña la función de soporte, y si nadie cuida del equipo, éste no se puede formar. Cada equipo necesita tener a alguien que se responsabilice de mantener la salud del equipo.

Los equipos también pueden tener falta de identidad si uno o más miembros estuvieran trabajando de forma individual en vez de formar parte del equipo. Hay muchos sitios apropiados para personas que trabajan de esta forma, pero su presencia puede ser fatal para la formación de un equipo.

Falta de reconocimiento. Algunas veces los miembros de un proyecto participan en el equipo de un proyecto en el que ponen su corazón y su alma, y sólo sienten que su esfuerzo no se aprecia. Si una organización desea crear un equipo de gran rendimiento en más de una ocasión, debería asegurarse de reconocer el esfuerzo extraordinario la primera vez de forma apropiada.

Barreras de la productividad. A Veces los equipos fallan porque creen que no pueden ser productivos. Los equipos no pueden sobrevivir si se les impide realizar su trabajo. Algunos expertos dicen que la primera función de un director de software es eliminar las barreras de la productividad, de forma que los desarrolladores que estén motivados por sí mismos puedan ser productivos [De Marco y Lister, 1987]

Comunicación ineficiente. Los equipos no se forman si no se pueden comunicar regularmente.

Falta de confianza. La falta de confianza puede destrozar la moral de un equipo. Una razón por la que normalmente los equipos no se forman dentro de organizaciones burocráticas es porque las organizaciones (en diversa medida) están basadas en la falta de confianza. La falta de confianza en los empleados suele estar institucionalizada.

Los directivos que prestan más atención a los detalles de administración del equipo que a los resultados que consiguen, demuestran una falta de confianza. Los directivos que gestionan poco las actividades de sus equipos, que no les permiten reunirse con sus clientes o que les dan fechas límite falsas, están dando signo de que no confían en ellos.

Personas problemáticas. Algunos programadores intimidan a sus compañeros utilizando sus planteamientos de diseño. Sus compañeros, en vez de enfrentarse, se conforman con sus demandas de diseño, aumentando su contacto con ellos.

Si un responsable tolera a un solo desarrollador al que sus compañeros consideran como un problema, se estará influyendo en la moral de los buenos desarrolladores.

Las reclamaciones más intensas y consistentes de los miembros del equipo eran que los responsables del equipo no estaban dispuestos a enfrentarse y resolver los problemas asociados al mal cumplimiento individual por parte de algunos miembros del equipo [Larson y La Fasto, 1989]. Los desarrolladores consideran esta conducta del responsable como un punto negro muy significativo de la directiva, ya que los directivos casi siempre piensan que sus equipos van mejor de lo que creen sus miembros.

Los problemas personales suelen ser fáciles de identificar si se sabe dónde buscarlos:

- Ocultan su ignorancia en vez de intentar aprender de sus compañeros de equipo.
- Tienen un deseo excesivo de intimidad.
- Son reservados.
- Se quejan de las decisiones de los equipos y continúan revisando antiguas discusiones después de que el equipo haya terminado.
- Los otros miembros del equipo se ríen o se quejan de la misma persona. Los desarrolladores de software no suelen quejarse directamente, de forma que hay que preguntar si tienen problemas cuando se oye mucho ruido.
- No se ponen a trabajar en las actividades del equipo.

Enseñar a la persona problemática a trabajar en equipo a veces funciona, pero normalmente es mejor que lo haga el equipo en lugar de que intenten resolverlo los directivos o los responsables. Habría que preparar al equipo sobre cómo manejar al miembro problemático del equipo.

Si la preparación no produce resultados rápidamente, no tema echar a una persona que no tiene el menor interés por el equipo. He aquí las tres razones principales:

- Es raro ver un problema importante provocado por la falta de experiencia. Es casi siempre cuestión de actitud, y las actitudes son difíciles de cambiar.
- Cuanto más tiempo se mantiene una persona que causa problemas, esta persona adquirirá más legitimidad a través de contactos casuales con otros grupos y directivos, un crecimiento de la base del código que la persona tiene que mantener, etc.
- Algunos directivos comentan que nunca se han lamentado de expulsar a alguien. Sólo lamentan no haberlo hecho antes.

Un directivo se podría preocupar por la pérdida de la base si se sustituye a un miembro del equipo, pero en un proyecto de cualquier tamaño, se compensará la pérdida de base por la eliminación de una persona que esté trabajando en contra del resto del equipo. Hay que cortar la pérdida, y entonces la moral del resto del equipo mejorará.

5.5.- Configuración de equipos a largo plazo.

A continuación se muestran algunas de las razones por las que es necesario mantener unidos a los equipos estables.

Mayor productividad. Con una estrategia de equipo estable, un grupo se mantiene unido si se crea dentro de un equipo, y se disuelve si no lo hace. Si se mantienen unidos a los equipos productivos, se consiguen grandes ganancias. El efecto total que genera es una “media superior” del nivel de rendimiento en la organización.

Costes iniciales menores. Manteniendo juntos a los equipos efectivos, se mantiene parte de la visión, identidad del equipo, comunicación, confianza, y el depósito de buena voluntad derivados de la finalización de un buen proyecto juntos. También se mantienen los métodos técnicos específicos y el conocimiento de herramientas específicas dentro de un grupo.

Menor riesgo de problemas con el personal. Los problemas personales que surgen por personas que no trabajan bien juntas cuestan al proyecto tiempo y dinero. Estos problemas se pueden evitar completamente manteniendo al equipo unido una vez formado.

Menos cambios de personal. DeMarco y Lister estiman que el 20 por 100 de la medida del gasto total de la compañía se va en cambios de personal [DeMarco y Lister, 1987].

Resulta menos probable que las personas que se han integrado en grupos unidos abandonen la compañía que las que no lo han hecho [Lakhanpal, 1993]. Los desarrolladores han encontrado un entorno en el que se divierten y pueden sentirse productivos.

El tema de la inactividad. En ocasiones, las organizaciones se muestran recelosas de mantener a los equipos juntos, ya que tendrían que estar pagando a un equipo que no está haciendo nada hasta que llegue un proyecto apropiado para empezar a trabajar. Ésta es una objeción válida, pero la mayoría de organizaciones, en el fondo, no lo consideran últimamente como un fuerte inconveniente.

Las organizaciones que miran exclusivamente el coste del tiempo perdido no tienen en cuenta los costes de reconstrucción de los equipos para cada proyecto nuevo. El coste de construcción de un equipo nuevo incluye el coste de unir al equipo y el de intentar que el equipo funcione conjuntamente.

Las organizaciones tienden a no tener en cuenta el dinero que se pierde separando un equipo de alto rendimiento.

Algunas organizaciones se preocupan de que si se mantienen a los equipos juntos, no les será posible conseguir equipos para trabajar en ciertos proyectos. Pero otros han descubierto que si dan a la gente la oportunidad de trabajar con las personas que les gusten, trabajarán en cualquier proyecto [DeMarco y Lister, 1987].

Finalmente, aún está por ver una organización de desarrollos de software con grandes períodos de inactividad. Por el contrario, la mayoría de los proyectos comienzan tarde porque el personal no está disponible hasta que han finalizado otros proyectos anteriores.

Ahora sabemos que entre grupos de personas con características equivalentes, los equipos más productivos son 2 ó 3 veces como los equipos menos productivos. Son una y media o dos veces tan productivos como la media de los equipos. Si se tiene un equipo que se sabe está a la cabeza del rango, hay que ser suficientemente inteligente como para permitirles perder el tiempo, durante hasta un tercio e incluso la mitad de su vida labora, para evitar el riesgo de que el equipo se divida y tenga que sustituirlo simplemente por un equipo medio.

5.6.- Resumen de directrices para grupos de trabajo

Larson y La Fasto resumieron los resultados de sus investigaciones en un conjunto de directrices básicas para los responsables y miembros del equipo, directrices que refleja la siguiente tabla:

Responsable del equipo	Miembros del equipo
Como responsable del equipo, tendré que:	Como miembro del equipo, tendré que:
1. Evitar comprometer los objetivos del equipo con temas políticos.	1. Demostrar una comprensión realista de mi función y posibilidades.
2. Mostrar el compromiso personal con el objetivo del equipo.	2. Demostrar las opiniones objetivas y basadas en hechos.
3. No diluir los esfuerzos del equipo con demasiadas prioridades.	3. Colaborar eficientemente con los demás miembros del equipo.
4. Ser justo e imparcial con todos los miembros del equipo.	4. Poner los objetivos del equipo por encima de cualquier objetivo personal.
5. Estar dispuesto a hacer frente y resolver temas asociados con el rendimiento inadecuado por parte de los miembros del equipo.	5. Mostrar la voluntad de desarrollar el esfuerzo necesario para conseguir el éxito del equipo.
6. Estar abierto a nuevas ideas e información por parte de los miembros del equipo.	6. Estar dispuesto a compartir información, percepción y realimentación apropiadamente.
	7. Proporcionar ayuda a los demás miembros del equipo cuando la necesiten.
	8. Demostrar altos niveles de excelencia.
	9. Apoyar y admitir las decisiones del equipo.
	10. Demostrar que se es consecuente con sus principios, enfrentándose a temas importantes.
	11. Demostrar aspectos de responsabilidad de forma que contribuya al éxito del equipo.
	12. Responder constructivamente a la realimentación de los demás.

Fuente: adaptado de [Larson y La Fasto, 1989].

6.- Estructura del equipo

Aunque se tenga personal formado, motivado y acostumbrado a trabajar duro, una estructura errónea del equipo puede minar su esfuerzo en lugar de catapultarlo al éxito. Una mala estructura de equipo puede aumentar el tiempo de desarrollo, reducir la calidad, deteriorar la moral, incrementar los cambios de personal y llevar a la cancelación del proyecto. En la actualidad prácticamente un tercio de todos los equipos de proyectos están organizados de forma ineficaz [Jones, 1994].

Para estructurar equipos de proyecto e incrementar la productividad, se deben evitar las estructuras que producen fricciones. Así, un equipo de trabajo bien estructurado debe poder enfrentar la rotación personal del equipo y la falta de control del director sobre los recursos, favorecer la comunicación entre los miembros del equipo y facilitar la integración de las diversas partes entre sí.

No hay una sola estructura adecuada para todos los proyectos. La estructura que es ideal para un proyecto puede ser un fracaso para otro. Al configurar la estructura de un equipo es preciso tener en cuenta varias cosas: tamaño del proyecto, disponibilidad del personal, características culturales de la empresa y las características psicológicas de los miembros del equipo y otras personas involucradas (capítulo 5).

6.1.- Consideraciones sobre la estructura del equipo

La primera consideración al organizar un equipo es determinar el objetivo general del mismo. Según Larson y La Fasto [1989] estos son los objetivos generales:

- Resolución de problemas.
- Creatividad.
- Ejecución táctica.

Una vez establecido el objetivo general, hay que seleccionar una estructura de equipo que se adapte al mismo. La estructura más apropiada dependerá del objetivo del equipo.

6.1.1.- TIPOS DE EQUIPOS

Una vez identificado el objetivo más general del equipo, se prepara una estructura de equipo que haga hincapié en la característica más importante para este tipo de equipo.

Equipo para la resolución de problemas. El equipo para resolución de problemas se centra en resolver un problema complejo, poco definido.

Los miembros de un equipo de resolución de problemas tienen que ser de confianza, inteligentes y pragmáticos. Los equipos de resolución de problemas están ocupados principalmente en una o más cuestiones específicas y su estructura de equipo debe soportar este modo de trabajo.

Equipo para creatividad. El objetivo de un equipo para creatividad consiste en explorar posibilidades y alternativas.

Los miembros de un equipo para creatividad necesitan estar motivados, ser independientes, creativos y persistentes. La estructura del equipo necesita soportar la autonomía individual y colectiva de los miembros del equipo.

Equipo para ejecución táctica. Se centran en ejecutar un plan bien definido. Este tipo de equipo se caracteriza por tener tareas muy centradas y funciones muy definidas. Los criterios del éxito tienden a ser todo o nada, así que a menudo resulta fácil decir si el equipo triunfa o fracasa.

Los miembros de un equipo de ejecución táctica necesitan tener un sentido de la urgencia de su misión, estar más interesados en la acción que en la intelectualización esotérica y ser leales al equipo.

La siguiente tabla resume los diferentes objetivos de los equipos y las estructuras de equipo que soportan dichos objetivos:

	Objetivo general		
	Resolución de problemas	Creatividad	Ejecución táctica
<i>Característica dominante</i>	Confianza	Autonomía	Claridad
<i>Ejemplo típico en software</i>	Mantenimiento corrector en sistemas en marcha	Desarrollo de un nuevo producto	Desarrollo de la actualización de un producto
<i>Énfasis en el proceso</i>	Centrado en cuestiones	Explorar posibilidades y alternativas	Tareas muy centradas con funciones claras, marcadas con un claro éxito o fracaso
<i>Modelos apropiados de ciclo de vida</i>	Codificar y corregir, espiral	Prototipado evolutivo, entrega evolutiva, espiral, diseño para planificación, diseño para herramientas	Cascada, cascadas modificadas, entrega por etapas, espiral, diseño para planificación, diseño para herramientas
<i>Criterios de selección del equipo</i>	Inteligentes, desenvueltos, sensibles, alta integridad	Cerebrales, independientes, pensadores, con iniciativa, tenaces	Leales, comprometidos, dispuestos a la acción, con sentido de la urgencia, responsables
<i>Modelos apropiados de equipo de software</i>	Equipo de negocios, equipo de búsqueda y rescate, equipo SWAT	Equipo de negocios, equipo con programador jefe, equipo en la sombra, equipo de prestaciones, equipo de teatro	Equipo de negocios, equipo con programador jefe, equipo de prestaciones, equipo SWAT, equipo profesional de atletismo

Fuente: [Larson y La Fasto, 1989].

6.1.2.- CARACTERÍSTICAS ADICIONALES PARA DISEÑO DE EQUIPOS

Aparte de los tres tipos básicos de equipos, hay cuatro características de la estructura de los equipos que parecen caracterizar todos los tipos de equipos que funcionan con efectividad.

Papeles y responsabilidades claros. En un equipo de alto rendimiento, todo el mundo cuenta, y cada uno sabe lo que tiene que hacer.

Monitorización del rendimiento individual y realimentación. La otra cara de las responsabilidades consiste en que los miembros del equipo necesitan saber si están trabajando de acuerdo con las expectativas del equipo. El equipo necesita disponer de mecanismos para que sus miembros sepan en qué aspectos tienen un rendimiento aceptable y cuáles son los que necesitan mejorar.

Comunicación efectiva. La información debe ser fácil de consultar. Reservar la información y darla a conocer “cuando haga falta” es malo para la moral en un proyecto. Se debe poner toda la información relevante a disposición de todos.

La información tiene que venir de fuentes fidedignas. La confianza del equipo en la toma de decisiones depende de lo fiable que sea la información sobre la que basa sus decisiones.

Los miembros del equipo necesitan tener oportunidades para plantear cuestiones que no se encuentren en la agenda formal, esto forma parte del éxito de métodos de gestión informales tales como *walking around* (“gestión mediante visitas”).

El sistema de comunicación tiene que reflejar las cuestiones planteadas y las decisiones tomadas. Mantener un registro minucioso evita que el equipo retroceda para plantearse decisiones pasadas.

Toma de decisiones basada en hechos. Los juicios subjetivos pueden minar la moral del equipo. Los miembros de un equipo de alto rendimiento necesitan comprender las bases de todas las decisiones que les afectan. Si encuentran que las decisiones están tomadas por razones arbitrarias, subjetivas o egoístas su rendimiento se verá penalizado.

6.1.3.- ¿CUÁL ES EL MEJOR TIPO DE EQUIPO PARA UN DESARROLLO RÁPIDO?

Una de las claves para organizar un equipo de desarrollo es comprender que no existe una única estructura de equipo que alcance la máxima velocidad de desarrollo en cualquier proyecto. No existe una única “estructura de equipo para desarrollo rápido”, ya que la estructura más eficaz depende del contexto.

6.2.- Modelos de equipo

Algunos de los modelos sólo afectan al modo interno de funcionamiento del equipo, y pueden ser implementados por el responsable técnico o el propio equipo. Otros afectan a la relación entre el equipo y la directiva, y generalmente requerirán la aprobación por parte de la misma.

6.2.1.- EQUIPO DE NEGOCIOS

La estructura de equipo más común es la de un grupo de iguales encabezado por un jefe técnico. Aparte del jefe técnico, todos los miembros del equipo tienen el mismo estatus, diferenciándose en su ámbito de experiencia. El jefe técnico contribuye de forma activa.

El jefe es responsable de tomar las decisiones finales sobre cuestiones técnicas.

Desde el exterior, la estructura del equipo de negocios parece una estructura jerárquica típica. Concentra la comunicación con la directiva, identificando una persona como responsable principal del trabajo técnico del proyecto. Permite a cada miembro del equipo trabajar en su área de experiencia, y permite que el propio equipo decida en que va a trabajar cada uno de sus miembros. Funciona bien con grupos pequeños y con grupos que llevan juntos mucho tiempo, y pueden estructurar sus relaciones con el tiempo.

Es bastante adaptable para trabajar en todos los tipos de proyectos. Pero generalmente ésta es su debilidad, y en muchos casos hay otra estructura que puede funcionar mejor.

6.2.2.- EQUIPO CON PROGRAMADOR JEFE O QUIRÚRGICO

Esta idea de equipo fue concebida por IBM a finales de los sesenta y popularizada por Fred Brooks [1975, 1995], él se refiere al equipo como quirúrgico, ambos términos son intercambiables.

El equipo con programador jefe saca partido del hecho de que algunos desarrolladores son 10 veces más productivos que otros.

En el concepto de equipo quirúrgico, la persona más cualificada se identifica con el cirujano, o programador jefe. Esta persona realiza la especificación completa, hace todo el diseño, escribe la mayoría del código de producción, y es el responsable final de prácticamente todas las decisiones del proyecto. El resto de los miembros del equipo son libres para especializarse.

Los miembros del equipo se distribuyen alrededor del cirujano con funciones de apoyo, y el equipo con programador jefe saca partido del hecho de que los especialistas tienden a rendir más que los generalistas [Jones, 1991].

Hay un programador de reserva que apoya al cirujano como crítico, ayudante de investigación, contacto técnico con grupos externos y cirujano de reserva. Otra persona gestiona cuestiones administrativas. Aunque el cirujano tiene la última palabra en estas cuestiones, el administrador libera al cirujano de tener que tratarlas de forma diaria. Alguien será el responsable de crear herramientas personalizadas solicitadas por el programador jefe. El equipo está arropado por un “especialista del lenguaje”, que apoya al cirujano respondiendo a preguntas específicas y peculiares sobre el lenguaje de programación que está usando el programador jefe.

Varios de los papeles de apoyo sugeridos en la propuesta original de equipo con programador jefe son desempeñados actualmente por personas que no son programadores.

Esta estructura sigue siendo adecuada cuando se usa de forma oportunista. Este equipo resulta adecuado para proyectos creativos, en los que tener un cerebro al frente ayudará a proteger la integridad conceptual del sistema. También resulta adecuado para proyectos de ejecución táctica, en los que el programador jefe puede ser una especie de dictador que diseñe los medios más expeditivos para lograr la culminación del proyecto.

6.2.3.- EQUIPO EN LA SOMBRA

Un equipo en la sombra (*skunworks*) es una parte integral de la herencia del mundo de la ingeniería. Un equipo en la sombra aglutina un grupo de desarrolladores de productos creativos y con talento, les pone en una situación en la que son liberados de las restricciones burocráticas habituales de la organización, y les da libertad para desarrollar e innovar.

Los equipos en la sombra son tratados generalmente como cajas negras por sus directivos. La directiva no quiere conocer los detalles sobre como realizan el trabajo, sino que sólo quieren saber lo que están haciendo. Así, el equipo es libre de organizarse como mejor le parezca. Con el paso del tiempo, puede aparecer un líder natural, o el equipo podría designar uno desde el principio.

Los proyectos en la sombra tienen la ventaja de crear una sensación de intensa propiedad y compromiso por parte de los desarrolladores implicados. El efecto motivacional puede ser impresionante. Tienen la desventaja de no ofrecer mucha visibilidad del progreso del equipo. Parte de esto puede ser un efecto inevitable de la impredecibilidad inherente en un trabajo altamente creativo. También se debe a un equilibrio explícito: pérdida de visibilidad a costa de incrementar la motivación.

Los equipos en la sombra resultan más adecuados para proyectos exploratorios en los que la creatividad es absolutamente importante. Los equipos en la sombra son raramente la estructura más rápida cuando se necesita resolver un problema claramente definido o cuando se necesita ejecutar un plan bien entendido.

6.2.4.- EQUIPO DE PRESTACIONES

En un equipo de prestaciones, el desarrollo, el control de calidad, la documentación, la gestión del programa y el marketing están organizados con las estructuras jerárquicas tradicionales de responsabilidad.

Con esta organización tradicional se encuentran los equipos que toman uno o más miembros de cada uno de los grupos implicados y les asignan la responsabilidad de una parte de la funcionalidad del producto [Mc Carthy, 1995].

Los equipos de prestaciones presentan las ventajas de potenciación, facilidad de seguimiento y equilibrio. El equipo puede potenciarse sensiblemente porque incluye representantes de desarrollo, control de calidad, documentación, gestión del programa y marketing; es decir, de todas las partes implicadas. El equipo considerará todos los puntos de vista necesarios en sus decisiones, y por ello en raras ocasiones dará pie a que estas sean cuestionadas. Por la misma razón, resulta fácil llevar el seguimiento del equipo. Tienen acceso a todas las personas que necesitan para tomar buenas decisiones. Si no toman buenas decisiones, sólo pueden echarse la culpa a ellos mismos. El equipo está equilibrado.

Los equipos de prestaciones resultan adecuados para proyectos de resolución de problemas, ya que ofrecen el refuerzo y la facilidad de seguimiento necesarios para resolver las cuestiones planteadas de forma expeditiva. También son adecuados para proyectos de creatividad, ya que la composición interdisciplinar del equipo puede estimular las ideas. La gestión adicional generada por los equipos de prestaciones se desperdicia en los proyectos de ejecución táctica: si todas las tareas están claramente identificadas, los equipos de prestaciones tienen poco que aportar.

6.2.5.- EQUIPO DE EMERGENCIAS O DE BUSQUEDA Y RESCATE.

Este tipo de equipo se centra en resolver un problema específico. Hay que combinar un conocimiento especializado de herramientas software y hardware específico con un conocimiento igualmente especializado sobre un entorno comercial determinado.

Este equipo necesita un conocimiento íntimo del terreno en el que va a buscar, la capacidad de responder inmediatamente, y un excelente conocimiento sobre el modo de estabilizar el sistema en poco tiempo, controlando el problema de inmediato.

Esta estructura de equipo es la más apropiada para equipos que necesitan centrarse en la resolución de un problema. Está demasiado dirigido a problemas básicos como para soportar mucha creatividad, y está demasiado orientado a corto plazo como para soportar la ejecución táctica.

6.2.6.- EQUIPO DE ESPECIALISTAS (G.E.O.)

En este tipo de equipo, cada miembro está especialmente entrenado en algún tema. Los equipos se “entrenan” constantemente, para que cuando estalle una crisis puedan trabajar conjuntamente como un bloque sin fisuras.

La idea básica es coger un grupo de personas con una sólida formación en una herramienta o método determinado, y dedicarlas a un problema que se adapta perfectamente a ser resuelto con esta herramienta o método.

Generalmente son equipos permanentes, están habituados a trabajar juntos, y tienen papeles bien definidos.

Resultan especialmente adecuados en proyectos de ejecución táctica. Su trabajo no consiste en ser creativos, sino en implementar una solución dentro de los límites de una herramienta o metodología que conocen perfectamente. También funcionan bien en proyectos de resolución de problemas. Los miembros del equipo tienen confianza mutua, y su concentración en una fase determinada del proyecto les permite tratar la realización de esa fase como un único problema que pueden superar rápidamente.

6.2.7.- EQUIPO DEPORTIVO

Los desarrolladores son seleccionados como mínimo tan cuidadosamente como los directivos, y probablemente resultan más críticos para el éxito del proyecto.

El presidente de un equipo deportivo gestiona las decisiones, entre bastidores, que son importantes estratégicamente pero no es el que golpea la bola.

El responsable de software también es importante, pero no por ninguna de las capacidades de desarrollo. El papel del responsable es despejar el camino de obstáculos y permitir que los desarrolladores trabajen de forma eficiente. Los desarrolladores podrían desarrollar un producto sin el responsable, pero el responsable no podría sin los desarrolladores. No tiene sentido ver al responsable como una persona mejor o que está jerárquicamente por encima de los desarrolladores.

Los equipos deportivos también tienen papeles altamente especializados. El responsable del proyecto puede contratar a un especialista, por ejemplo en bases de datos, pero no debe esperar de él otra cosa.

Este modelo específico se aplica mejor a los proyectos de ejecución táctica, que hacen énfasis en los papeles altamente especializados que desarrollan las personas individuales.

6.2.8.- EQUIPO DE TEATRO

Se caracteriza por una fuerte dirección y una gran negociación de los papeles del proyecto. El papel central del proyecto es ocupado por el director, que mantiene la visión del producto y asigna a la gente responsabilidades en áreas individuales. Los participantes individuales pueden adaptar sus papeles, sus partes del proyecto, como les dicten sus propios instintos artísticos. Pero no pueden llevar demasiado lejos sus propias ideas, evitando colisionar con la visión del director. Si sus ideas chocan con la visión del director, la visión del director tiene que prevalecer para la buena marcha del proyecto.

En el modelo de teatro, no se está asignado simplemente a un proyecto. Se participa en una audición, y se acepta un papel. Antes de poder aceptar un papel hay una larga negociación.

La fuerza del modelo de teatro consiste en que proporciona una vía para integrar importantes contribuciones individuales junto con una fuerte visión central en proyectos de creatividad. Como afirma Fred Brooks, la integridad conceptual es la consideración más importante en el diseño de un sistema, y si un sistema necesita tenerla, hay una persona que tiene que controlar los conceptos [Brooks, 1975].

El modelo de teatro resulta particularmente apropiado para equipos software dominados por fuertes personalidades.

Es un modelo adecuado para proyectos multimedia modernos. Mientras anteriormente los proyectos software necesitaban integrar las contribuciones de varios desarrolladores de software, ahora tienen que integrar las contribuciones de varias personas pertenecientes a distintas especialidades.

6.3.- *Grandes equipos*

Los equipos grandes plantean problemas especiales de comunicación que se comentan más adelante y coordinación.

Los proyectos grandes requieren métodos de organización que formalicen y simplifiquen las comunicaciones. La formalización de las comunicaciones es un elemento importante para tener éxito en los grandes proyectos. La simplificación de las comunicaciones puede resultar muy afectada por la estructura del equipo.

Todos los métodos para simplificar las comunicaciones se basan en la creación de algún tipo de jerarquía, es decir, crear grupos pequeños que funcionen como equipos, y luego asignar responsables dentro de dichos grupos para interactuar entre sí y con la directiva. Se pueden crear estos grupos reducidos de varias formas:

- Formar un conjunto de equipos de negocios, y luego designar un nexo en cada equipo para comunicarse con los otros grupos.
- Formar un conjunto de equipos con jefe programador, y hacer responsable de la comunicación con otros grupos al programador de reserva.
- Formar equipos de prestaciones, y hacer responsable de la comunicación con otros grupos al representante de la gestión de programas.

Independientemente de la organización de los equipos pequeños, resulta crítico que haya una única persona como responsable final de la integridad conceptual del producto. Esta persona puede tener el papel de arquitecto, cirujano, director, o incluso el responsable del programa, pero tiene que ser una persona cuyo trabajo consista en asegurarse de que todas las buenas soluciones locales de los equipos conforman una buena solución global.

6.4- Mala comunicación

Los equipos plantean, los grandes más, problemas de comunicación y coordinación. La información se retiene, se distorsiona, etc. Sirviendo de lastre más que de motor al desarrollo.

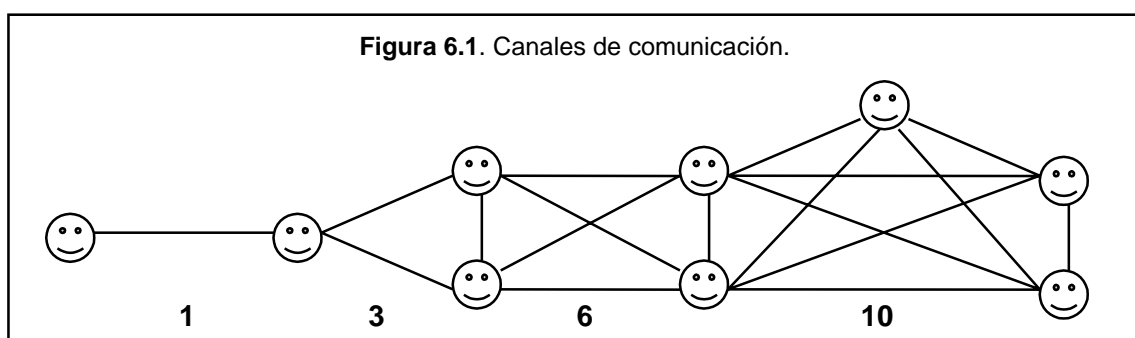
La información es un elemento vital de los proyectos; y comunicar esa información eficazmente a las partes interesadas es de suma importancia para alcanzar el éxito. Cuando la comunicación se interrumpe, el proyecto corre serio riesgo.

Examinaremos ahora tres de las diversas clases de fricción basadas en la comunicación.

6.4.1.- LA COMUNICACIÓN COMO FIN Y NO COMO MEDIO

A medida que los proyectos se burocratizan, se invierte más y más se esfuerza en transmitir información y coordinar tareas. En los proyectos grandes, ese trabajo puede igualar al de la realización del proyecto en sí.

Como se puede apreciar en la figura 6.1., el número de canales de comunicación no se incrementa de forma lineal, a medida que aumenta el número de participantes. Se incrementa de forma exponencial, proporcional al *cuadrado* del número de personas. En términos matemáticos, si el equipo está compuesto por n miembros, hay potencialmente $n(n-1)/2$ canales que mantener.



Por supuesto, en la mayoría de los proyectos nadie tiene necesidad de comunicarse con nadie. No obstante, existe la posibilidad de abrumar al equipo con exigencias de comunicación. Cuando se dedica demasiado tiempo a enviar y recibir mensajes, la productividad del equipo empieza a disminuir.

6.4.2.- ARTERIOESCLEROSIS DE LA INFORMACIÓN

La arterioesclerosis de la información se produce cuando los canales de comunicación están tan obstruidos que la información importante apenas si puede circular.

La obstrucción es, en gran parte, consecuencia de la exigencia de que la información sea procesada de manera burocrática. Puede ser causada también por grandes cantidades de información inútil que circula por los canales y bloquea el flujo de mensajes importantes.

La principal consecuencia de la arterioesclerosis informática es que el flujo de la información se demora y el trabajo del equipo se ve perjudicado.

6.4.3.- MENSAJES DISTORSIONADOS

Con frecuencia en las organizaciones: los mensajes se distorsionan. Las consecuencias de los mensajes distorsionados son diversas, desde inocuas hasta catastróficas. En los proyectos, las instrucciones que llegan a destino distorsionadas pueden lograr que el equipo realice mal su trabajo. Si ese trabajo tiene que hacerse nuevamente (suponiendo que se detecte el error a tiempo) o si causa problemas con otras tareas, la productividad del equipo disminuye.

7.- Conclusiones

Está claro que todo lo relacionado con *peopleware* influye fuertemente en la productividad, queda muy claro que cualquier organización que trate seriamente de mejorar la productividad primero debe ocuparse de temas relacionados con personal, como la motivación, equipo de trabajo, selección de personal y formación. Hay otras formas de mejorar la productividad, pero la gestión de personal ofrece los mayores beneficios potenciales.

Hay que usar poco y buen personal, asignar tareas según la habilidad y motivación de la gente disponible, ayudar a la gente a actualizarse por sí misma en vez de obligarles a trabajar donde más experiencia tienen o donde son más necesarios, seleccionar a gente que se complemente y armonice con los demás. eliminar y reemplazar a los miembros problemáticos del equipo lo antes posible.

La forma de organizar al personal tiene un gran efecto sobre la eficiencia con la que trabajen. Hay que sacar partido a la estructura de los equipos para que concuerden con el tamaño del proyecto, las características del producto y los objetivos de planificación. Un proyecto software específico también puede sacar provecho de la especialización apropiada.

Una persona que carece de motivación no va a querer trabajar duro, y prefiere dejarse llevar. La motivación es el único factor que provocará que una persona renuncie a las tardes y a los fines de semana sin que se le pida. Pocos otros factores pueden aplicarse a tanta gente dentro de tantos equipos en tantas empresas. La motivación es potencialmente el aliado más fuerte que tenemos.

Hay que evitar caer una y otra vez en los mismos errores y aprender de ellos.

Estudio tras estudio se ha demostrado que la motivación probablemente tiene mayor efecto sobre la productividad y la calidad que ningún otro factor. La capacidad individual de los miembros del equipo, así como sus relaciones como equipo, también tienen una influencia sobre la productividad. Medir el trato con el equipo de desarrollo, especialmente con empleados problemáticos y evitar las hazañas y comportamientos heroicos ya que impiden la cooperación entre los múltiples elementos que contribuyen al proceso de desarrollo del software, de igual manera unas expectativas poco realistas contribuirán a la percepción de que el plan de desarrollo es demasiado largo, y de que puede ser malo.

El entorno será otra de las variable importantes a tener en cuenta. La mayoría de los desarrolladores consideran sus condiciones de trabajo como insatisfactorias. Los trabajadores que están en oficinas silenciosas y privadas tienden a funcionar significativamente mejor que aquellos que ocupan cubículos en salas ruidosas y repletas. Los entornos repletos y ruidosos alargan los planes de desarrollo.

Los conflictos de personalidad crean un mal ambiente cuyo principal efecto es la mala comunicación, y los efectos secundarios de la mala comunicación incluyen el pobre entendimiento de los requerimientos, pobre diseño de la interfaz de usuario y, en el peor caso, el rechazo del cliente a aceptar el producto acabado. En la mayoría de los casos suelen ser conflictos de personalidad entre dos grupos.

Hay que fomentar la participación, sino los principales participantes del esfuerzo de desarrollo de software no se implicaran en el proyecto. La cooperación

estrecha solo se produce si se han implicado todos los participantes, permitiendo una coordinación precisa del esfuerzo para un desarrollo rápido y satisfactorio, que es imposible conseguir sin una buena participación. Haciendo hincapié en la participación del usuario; los proyectos que no implican al usuario desde el principio corren el riesgo de que no se comprendan los requisitos del proyecto, y son vulnerables a que se consuma tiempo en prestaciones que más tarde retrasaran el proyecto.

Otro fallo muy común es primar la política en vez de los resultados, lo que será fatal para el desarrollo.

En lo referente a selección y formación, se podrían considerar como tareas separadas a la gestión de los recursos humanos, el éxito o fracaso de las mismas afectará a nuestro proyecto casi en la misma medida que una buena o mala gestión del personal. La participación directa de los equipos de dirección y gestión de proyectos es de suma importancia para el éxito en la adquisición de personal y asegura que los perfiles de los candidatos y las personas seleccionadas, se ajustan y tienen las características necesarias el proyecto y por ende el buen desarrollo del mismo.

Las personas son el meollo de los proyectos. El éxito radica en las personas. ¿Los miembros del equipo están comprometidos con el proyecto? ¿Son inteligentes? ¿Despliegan iniciativa en caso necesario? ¿Los jefes son solidarios? ¿Son explícitos con lo que esperan del personal? ¿Hay unas buenas relaciones con los clientes? Las respuestas a estas preguntas, y a otras afines, nos dan una idea de cómo actuar en nuestros proyectos.

Todos sabemos que las personas son multidimensionales, que la excesiva autocracia por parte de los directivos llevará a la desmoralización del personal creativo, y que se puede agrupar a las personas por “tipos”. Sin embargo, cuando actuamos como gerentes, solemos tratar a las personas como si fuesen unidimensionales, como si para ellas lo único importante fuera el trabajo. Podemos también ejercitar un estilo autocrático de management, y sentirnos amenazados por la realimentación legítima, que percibimos como una crítica injustificada a nuestro comportamiento. Y por otra parte, con frecuencia encasillamos a la gente en tipos simplistas: inteligente o estúpido, solidario o egoísta, bueno o malo.

Así, a pesar de lo obvias que son estas cuestiones en las relaciones humanas, una y otra vez las dejamos de lado en nuestra práctica gerencial. Es conveniente, por lo tanto, revisarlas cuidadosamente y tenerlas siempre presentes.

La planificación del proyecto será un factor definitorio de la raíz de los problemas en la gestión de recursos humanos. El método de planificación excesivamente optimista va contra la planificación efectiva, se inmiscuye en la vida personal del desarrollador, desgasta las relaciones con los clientes, contribuye a aumentar el cambio de personal, contribuye a bajar la calidad y daña a la empresa, impidiendo un progreso profesional y dando la sensación de que los desarrolladores de software no pueden cumplir lo que prometen.

Una planificación excesivamente optimista simplemente no funciona, no produce planificaciones más cortas sino todo lo contrario.

La planificación real más corta es el resultado de la planificación más precisa posible. Un proyecto que tiene problemas de planificación sólo necesita buscar la fuente de sus problemas en una planificación inicial irrealizable.

Una planificación excesivamente optimista predispone a que el proyecto falle por infravalorar el alcance del proyecto, minando la planificación efectiva, y reduciendo las actividades críticas para el desarrollo, como el análisis de requisitos o el diseño. También supone una presión excesiva para los desarrolladores, quienes a largo plazo se ven afectados en su moral y en su productividad.

Además de evitar una presión excesiva en la planificación derivada de un planificación excesivamente optimista, hay que seguir la planificación establecida, manteniendo siempre un nivel aceptable de presión para evitar una relajación en la ejecución del plan ya que los equipos de desarrollo hacen planes y rutinariamente los abandonan cuando se tropiezan con un problema en la planificación. Hay que buscar alternativas a los fallos en la planificación.

El efecto de una planificación puede ser el éxito o una destrucción de la moral del equipo y una motivación cero.

La motivación es indudablemente la mayor influencia individual sobre como trabajan las personas. La mayoría de los estudios de productividad han encontrado que tiene mayor influencia en la productividad que cualquier otro factor. La motivación juega un papel crítico en el desarrollo, pero es un factor intangible: es difícil de cuantificar, y suele considerarse después de otros factores menos importantes, pero más fáciles de medir. Cualquiera sabe que la motivación es importante, pero pocos hacen algo al respecto.

Hay que potenciar la realización de los desarrolladores, darles oportunidades de superación, intentar no inmiscuirse en sus vidas personales, todo ello es motivador. Es de vital importancia cuando se habla de motivación tener en cuenta que un desarrollador a veces no necesita más motivación que el trabajo en sí, si se explora está realidad adecuadamente, la directiva tendrá la seguridad casi al 100 por 100 de contar con un equipo motivado, que reaccionara rápidamente y sin objeciones a los retos planteados por el proyecto.

Todo ello siempre que se trate con respeto al equipo y se le respalde oficialmente no solo en los buenos momento.

Premios, incentivos y revisiones del rendimiento son otros factores que ayudan a motivar, y lo que es más importante, a mantener motivado a un equipo de desarrollo durante un proyecto.

El entorno de trabajo, los medios puestos a disposición del equipo, y la facilidad o dificultad de acceso a los mismos, actúan también como elementos o factores de motivación / desmotivación.

Hacer incidencia en factores potenciadores de la motivación no es suficiente si no se hace todo lo posible para evitar los factores que destruirán la motivación y la moral del equipo.

Los desarrolladores son susceptibles a ser manipulados por los directivos y lo que desean es que la directiva les trate de forma sincera.

Los desarrolladores responden igualmente mal a las razones dadas por la directiva de la empresa o el equipo de gestión del proyecto en lo referente a establecimiento de fechas límite, cambios en la planificación, etc. pueden parecer ser evasivas y manipuladoras.

Salvo circunstancias extremas cualquier directivo que pida a los desarrolladores que realicen trabajo extra en un proyecto les debe dar una clara explicación.

En ocasiones los directivos no ven la cantidad de trabajo que los desarrolladores están realizando, de forma que piensan que no están haciendo mucho y, por tanto, deben obligarlos. Si se desea que los desarrolladores hagan más de lo que realmente ponen de manifiesto en la oficina, nunca jamás hay que decirles que no están trabajando duro cuando si lo están haciendo.

Si no se involucra a los desarrolladores en decisiones que les afectan da la impresión de que el responsable no presta atención al equipo de desarrollo y la motivación caerá.

La motivación es necesaria pero como decía Sancho Panza “Ni tanto ni tan calvo, mi señor”.

Las campañas de motivación pueden resultar cargantes y hay algunos métodos de motivación con los que es más fácil insultar la inteligencia del desarrollador que motivarlo. Con los desarrolladores de software, funciona mejor un ligero toque.

Una vez motivados los desarrolladores hay que “unirlos”. Los equipos son la unidad de trabajo de los proyectos. Debido al importante papel que desempeñan los equipos en los proyectos, vale la pena dedicar cierto tiempo a estudiarlos. Una vez en posesión de este conocimiento se podrán estructurar equipos de proyecto que maximicen la probabilidad de que nuestro trabajo sea realizado con eficiencia.

Un equipo es un conjunto de personas que trabajan juntas para alcanzar determinado objetivo. Pero, sus esfuerzos deben ser coordinados.

Los equipos de proyecto, pueden tener diversas formas y tamaños. Algunos deben encarar los problemas de alta complejidad; otros, asuntos de rutina. Algunos proyectos son muy dinámicos y sus miembros cambian constantemente; otros son estables.

Estos últimos dos puntos tienen especial importancia. la presencia del espíritu de equipo puede dar a un grupo la competitividad que le permitirá imponerse a otros grupos, igualmente competentes, pero que carecen de esa vivencia de pertenecer a un equipo.

Evidentemente, una de las tareas más importantes de los gerentes de proyectos consiste en desarrollar en su plantel cierto sentido de identificación con el equipo.

Los equipos de proyecto tienen estructura, es decir, se organizan según reglas establecidas que rigen las relaciones de los miembros entre sí, con el director del proyecto, con el cliente y con el producto que se está desarrollando.

Un equipo bien estructurado puede incrementar las probabilidades de éxito, mientras que uno mal estructurado causará problemas y tal vez conduzca al fracaso. Por lo tanto, un equipo bien estructurado es una condición necesaria, pero no suficiente, para alcanzar el éxito. En cuanto al equipo mal estructurado, es una fórmula segura para fracasar.

Los proyectos se realizan por medio de equipos, pero muchas veces el equipo es fragmentario y está mal estructurado. El director, entonces, debe esforzarse por generar una estructura de equipo, creándola prácticamente de la nada. Y lo primero que hay que hacer es seleccionar una estructura adecuada para el equipo, una estructura que aumente la eficiencia.

Sin embargo, seleccionar una estructura adecuada no es simple. Es preciso alentar a los colaboradores para que se identifiquen con el equipo, desarrollen espíritu de cuerpo y hagan todo lo necesario para alcanzar el buen éxito del proyecto. El problema es que, por lo general los empleados no son permanentes y tienen poco interés en el esfuerzo emprendedor. Por lo tanto, el director debe ser capaz de generar un sentimiento de identidad en un medio que no favorece la cohesión: debe convertir a sus colaboradores en adeptos.

Los recursos humanos son potencialmente la mejor forma de potenciar la productividad, su gestión es decisiva en el éxito o fracaso de todo proyecto. Y esto en la mayoría de los casos se olvida o se obvia. Un directivo que no seleccione, forme, dirija de forma efectiva, motive y estructure adecuadamente a los equipos a su cargo, está abocado, antes o después (seguramente después, donde los costes aumentan) al fracaso. Hay muchos sitios apropiados para personas que trabajan de esta forma, pero su presencia es fatal para la buena marcha del proyecto.

Fuentes y Bibliografía

- ALTER, Steven (1992): **Information Systems. A Management Perspective**. Reading (Massachusetts), Addison-Wesley.
- ANDREU, Rafael; RICART, Joan y VALOR, Josep (1991): **Estrategia y Sistemas de Información**. Madrid, Mc Graw Hill.
- BASILI, Victor R. y MCGARRY, Frank (1995): **The Experience Factory: How to Build and Run One**, Tutorial M1, 17th International Conference on Software Engineering, Seattle, Washington, abril 24.
- BAYER, Sam y HIGHSMITH, Jim (1994): **RADical Software Development**. American Programmer, junio: 35-42
- BODDIE, John (1987): **The Crunch Mode**. New York, Yourdon Press
- BOEHM, Barry W. (1981): **Software Engineering Economics**. Englewood Cliffs, New Jersey, Prentice Hall.
- BOEHM, Barry W. (1987a): **Improving Software Productivity**. IEEE Computer, septiembre: 43-47.
- BOEHM, Barry W. (1987b): **Industrial Software Metrics**. Top10 List, IEEE Software, vol.4, núm.9 (septiembre): 84-85
- BOEHM, Barry W. y ROSS, Rony (1989): **Theory-W Software Project Management: Principles and Examples**. IEEE Transactions on Software Engineering SE-15 (julio): 902-916.
- BROOKS, Frederick P. Jr. (1975): **The Mythical Man-Month**. Reading, Massachusetts, Addison-Wesley.
- BROOKS, Frederick P. Jr. (1995): **The Mythical Man-Month, Anniversary Edition**. Reading, Massachusetts, Addison-Wesley.
- CERRATO, P. (2000): **Principios de Project Management**. Madrid, Institute for International Research.
- DAVISON FRAME, J. (1999): **La dirección de proyectos en las organizaciones**. Barcelona, Granica.
- DE MARCO, Tom y LISTER, Timothy (1985): **Programmer Performance and the Effects of the Workplace**, *Proceedings of the 8th International Conference on Software Engineering*. Washington, D.C.: IEEE Computer Society Press, 268-272.
- DE MARCO, Tom y LISTER, Timothy (1987): **Peopleware: Productive Projects and Team**. New York, Dorset House.
- DE PABLO, I. (1989): **El reto informático. La gestión de la información en la empresa**. Madrid, Pirámide.

- EMERY, Fred y EMERY, Merrelyn (1975): ***Participative Design-Work and Comunity***. Caanberra, Center for Continuing Education, Australian National University.
- FISHER, Roger y URI, William (1981): ***Getting to Yes***. New York, Penguin Books.
- GLASS, Robert L. (1994a): ***Software Creativity***. Englewood Cliffs, New Jersey, Prentice Hall PTR.
- GLASS, Robert L. (1994b): ***Object-Orientation is Least Successful Technology***. Sotfware Practitioner, enero: 1.
- GLASS, Robert L. (1994c): ***IS Field: Stress Up, Satisfaction Down***. Sotfware Practitioner, noviembre: 1,3.
- HACKMAN, J. Richard y OLDHAM, Greg R. (1980): *Work Redesign*. Reaading, Massachusets. Adisson-Wesley.
- HUMPHREY, Watts S. (1989): ***Managing the Software Process***. Reading, Massachusets, Addison-Wesley.
- INEM (1991): ***Gestión de los Recursos Humanos***. Madrid, Instituto Nacional de Empleo. Col. *Gestión empresarial para PYME*, Vol. 6.
- JONES, Capers (1991): ***Applied Software Measurement: Assuring Productivity and Quality***. New York, Mc Graw-Hill.
- JONES, Capers (1994): ***Assessment and Control of Software Risks***. Englewood Cliffs, New Jersey, Yourdon Press.
- KATZENBACH, Jon y SMITH, Douglas (1993): ***The Wisdom of Teams***. Boston, Harvard Business School Press.
- KIDDER, Tracy (1981): ***The Soul of a New Machine***. Boston, Atlantic Monthly/Little Brown.
- LAKHANPAL, B. (1993): ***Understanding the Factors Influencing the Performance of Software Development Groups: An Exploratory Group-Level Analysis***. Information and Software Technology, 35 (8): 468-473.
- LARSON, Karl E. y LA FASTO, Frank M.J. (1989): ***Teamwork: What Must Go Right; What Can Go Wrong***. Newbury Park, California, Sage.
- LYONS, Michael L. (1985): ***The DP Psyche***. Datamation, 15 de agosto: 103-109
- MAGUIRE, Steve (1994): ***Debugging the Development Process***. Redmon, Washington, Microsoft Press.
- MC CARTHY, Jim (1995): ***Dynamics of Software Development***. Redmond, Washington, Microsoft Press.
- MC CONNELL, Steve (1996): ***Desarrollo y Gestión de Proyectos Informáticos***. Madrid, Mc Graw Hill.
- PETERS, Tomás J. y WATERMAN, Robert H. Jr. (1982): ***In Search of Excellence***. New York, Warner Books.

- PIATTINI, Mario G.; CALVO-MANZANO, José A.; CERVERA, Joaquín y FERNÁNDEZ, Luis (1996): **Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión**. Madrid, RA-MA.
- PROJECT MANAGEMENT INSTITUTE (1996): **A Guide to the Project Management Body of Knowledge**. Upper Darby, Project Management Institute.
- RAE (1995): **Diccionario de la Lengua Española edición electrónica v.21.1.0**. Madrid. Espasa-Calpe.
- SENGE, Peter (1992): **La quinta disciplina**. Barcelona, Granica.
- STANDISH GROUP, The (1994): **Charting the Seas of Information Technology**. Dennis, Massachusets. The Standish Group.
- THOMSETT, Rob (1990): **Effective Project Teams: A Dilemma, A Model, A Solution**. American Programmer, julio-agosto: 25-35.
- VALLET, J. y MCGARRY, F. E. (1989): **A Summary of Software Measurement Experiences in the Software Engineering Laboratory**, Journal of Systems and Software, 9 (2): 137-148.
- VOSBURGH, J. B., et al. (1984): **Productivity Factors and Programming Environments**. *Proceedings of the 7th International Conference on Software Engineering*. Los Alamitos, California. IEEE Computer Society: 143-152.
- WEINBERG, Gerald M. (1971): **The Psychology of Computer Programming**. New York, Van Nostrand Reinhold.
- WEINBERG, Gerald M. (1994): **Quality Software Management, Volume 3: Congruent Action**. New York, Dorset House.
- ZAWACKI, Robert A. (1993): **Key Issues in Human Resources Management**, Information Systems Management, invierno: 72-75.

Fuentes en Internet

The Software Engineering Institute (SEI)

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
<http://www.sei.cmu.edu>

Society for Human Resource Management

Society for Human Resource Management
1800 Duke Street
Alexandria, Virginia 22314 USA
(703) 548-3440 FAX: (703) 836-0367
<http://www.shrm.org>

Management Centre Europe

Management Centre Europe
© Copyright 1999-2000
rue de l'Aqueduc 118
1050 Brussels
Belgium
+32 2 543 21 00
info@mce.be
<http://www.mce.be>

Myers-Briggs

<http://skeptdic.com/myersb.html> ¿En que consiste?
<http://elvis.rowan.edu/~cusumano/MBTest.html> Tipos psicológicos
<http://mijuno.larc.nasa.gov/dfc/mb.html> Explicación y tests
<http://www.geocities.com/Athens/Atrium/8202/pp/personparlour.html> Descripción de tipos y tests.

En www.cetus-links.org hay cientos de links a sitios web y páginas relativas a Project Management (entre otras cosas).

Anexo I. Métodos Recomendables

I.1.- Introducción a los métodos recomendables

Figura I.1. Tabla resumen de método recomendable.

Ejemplo de tabla resumen de método recomendable

Cada método recomendable tiene una tabla resumen que describe las características que definen dicho método.

Eficacia

Reducción potencial de la planificación nominal	Ninguna ($\approx 0\%$), Media (0-10%), Buena (10-20%), Muy buena (20-30%), Excelente (30%+)
Mejora en la visibilidad del progreso	Ninguna ($\approx 0\%$), Media (0-25%), Buena (25-50%), Muy buena (50-75%), Excelente (75%+)
Efecto sobre el riesgo de la planificación	Disminuye el riesgo, Sin efecto, Aumenta el riesgo
Posibilidad de éxito inicial	Baja ($\approx 0-20\%$), Media (20-40%), Buena (40-60%), Muy buena (60-80%), Excelente (80-100%)
Posibilidad de éxito a largo plazo	Baja ($\approx 0-20\%$), Media (20-40%), Buena (40-60%), Muy buena (60-80%), Excelente (80-100%)

Riesgos principales

- Esta sección resume los riesgos principales añadidos por este método *en el resto del proyecto*. No incluye los riesgos para tener éxito con el propio método.

Interacciones y equilibrios principales

- Esta sección resume las principales interacciones del método con otros métodos, y el equilibrio de factores implicados en el uso del método.

Fuente: adaptado de [McConnell, 1996]

Estos métodos representan el estado del arte actual dentro de la velocidad de desarrollo. Algunos de ellos simplemente parecen cuestiones de sentido común.

Todos los métodos recomendables están organizados de forma similar. Cada uno de ellos comienza con una tabla como la del ejemplo mostrado al inicio de este anexo.

La **reducción potencial de la planificación nominal**, contiene una estimación del efecto del uso del método sobre la planificación del proyecto. Todas las posibles reducciones suponen una ejecución del método de forma experta. Esta clasificación se realiza de forma verbal, la tabla de ejemplo de la figura I.1 muestra la correspondencia aproximada entre esta escala verbal y las reducciones en porcentaje.

La **mejora en la visibilidad del progreso** sería la mejor aproximación posible definiendo la mejora como el porcentaje del proyecto que un método hace visible respecto a un modelo de ciclo de vida tradicional en cascada. Esta clasificación también está dada en una escala verbal.

Efecto sobre el riesgo de la planificación. Indica si un método mejora la posibilidad de cumplir un plazo específico, no le afecta, o la empeora.

Posibilidad de éxito inicial. La dificultad del uso de cada método condiciona esta variable.

Posibilidad de éxito a largo plazo. Describe las posibilidades de que un método tenga éxito si se sigue fiel al mismo el tiempo suficiente para tener experiencia en su uso.

Riesgos principales. Esta entrada resume los riesgos principales que supone el uso de esta práctica sobre el resto del proyecto.

Interacciones y equilibrios principales. Describe la interacción del método con otros métodos, y los equilibrios de factores involucrados en el método.

Además de la tabla resumen, los métodos tienen prácticamente la misma organización, incluyendo las siguientes secciones:

- Uso del método recomendable.
- Gestión de los riesgos.
- Efectos secundarios.
- Interacciones del método recomendable con otros métodos
- Puntos cruciales.
- Claves para el éxito.
- Bibliografía adicional.

Cada método descrito en este anexo ha sido seleccionado por una de las siguientes razones:

- Reducción de las planificaciones de desarrollo.
- Reducción de las planificaciones de desarrollo percibidas (haciendo más visible el progreso).
- Reducción de la volatilidad de la planificación (reduciendo la oportunidad de que el proyecto se des controle).

1.2.- Entornos Productivos

Los entornos productivos ofrecen a los desarrolladores una protección frente al ruido y a las interrupciones necesaria para trabajar de forma efectiva. Además de mejoras en la productividad, algunas organizaciones han experimentado mejoras en la moral de los desarrolladores y en las tasas de retención después de implantar entornos productivos.

Eficacia

Reducción potencial de la planificación nominal:	Buena
Mejora en la visibilidad del progreso:	Ninguna
Efecto sobre el riesgo de la planificación:	Sin efecto
Posibilidad de éxito inicial:	Buena
Posibilidad de éxito a largo plazo:	Muy buena

Riesgos principales

- Pérdidas de productividad por mejoras en los despachos orientadas al estatus.
- Tiempo de transición.
- Repercusiones políticas del trato preferente a los profesionales del software.

Interacciones y equilibrios principales

- Requiere un pequeño incremento en el coste a cambio de un gran aumento de la productividad.

Más del 70 por 100 de todas las organizaciones de software trabajan en despachos aglomerados, y el tiempo medio entre interrupciones en esas condiciones es de sólo unos 11 minutos [Jones, 1994].

Las tareas de desarrollo de software requieren largos períodos de concentración ininterrumpida. Generalmente, los desarrolladores están muy motivados, y lo que realmente piden son unas condiciones que les permitan trabajar eficientemente.

Tiempo de concentración. Como cualquier actividad conceptual, la calidad del trabajo depende de la capacidad del trabajador para mantener un “estado de fluidez”: un estado relajado de plena inmersión en un problema que facilita la comprensión del mismo y la generación de soluciones [DeMarco y Lister, 1987]. Los desarrolladores trabajan mejor durante las horas en las que se encuentran en este estado de concentración sin esfuerzo. Requieren 15 minutos o más para entrar en un estado de fluidez, que puede perdurar entonces durante varias horas. Si son interrumpidos cada 11 minutos, posiblemente no entraran nunca en un estado de fluidez, y por tanto, difícilmente alcanzarán sus niveles de productividad.

Necesidades de higiene. Un despacho adecuado no incrementa la motivación o la productividad, pero por debajo de un cierto umbral, un espacio de trabajo inadecuado puede erosionar seriamente la motivación o la productividad. Una organización que no ofrece un entorno en el que se pueda trabajar efectivamente, no está sentando las bases que sus desarrolladores necesitan para ser productivos, y los desarrolladores tienden a ver estas organizaciones como si estuvieran trabajando irracionalmente contra sus propios intereses.

I.2.1.- USO DE ENTORNOS PRODUCTIVOS

Los entornos productivos presentan las siguientes características:

- Un espacio mínimo aproximado de 9 metros cuadrados por desarrollador.
- Un espacio de trabajo mayor de 1'5 metros cuadrados para poder tener libros, carpetas, notas, etcétera sobre el mismo simultáneamente. Los soportes de las mesas deben estar diseñados para no obstaculizar los movimientos de las piernas.
- Algún medio para detener las interrupciones telefónicas. La recomendación del uso del correo electrónico en lugar de llamadas telefónicas dentro de una organización es una de las medidas para prevenir las interrupciones telefónicas.
- Algún medio para detener las interrupciones en persona. Un despacho individual con una puerta es la solución más común. La política de la organización debe permitir y animar a los desarrolladores cerrar sus puertas; políticas de “puertas abiertas” perjudican la productividad del desarrollador.
- Algún medio de eliminar el ruido no deseado. Las conversaciones sociales y de negocios no deben tener lugar en el área de trabajo de los desarrolladores, y el sistema de megafonía no debe interrumpir para anunciar nada que sea menos serio que un incendio en el edificio. De nuevo, un despacho privado con puerta es la solución más habitual.
- Un mínimo de 5 metros lineales de estanterías.

Además de estos factores, presentes en informes publicados hay otros factores adicionales importantes:

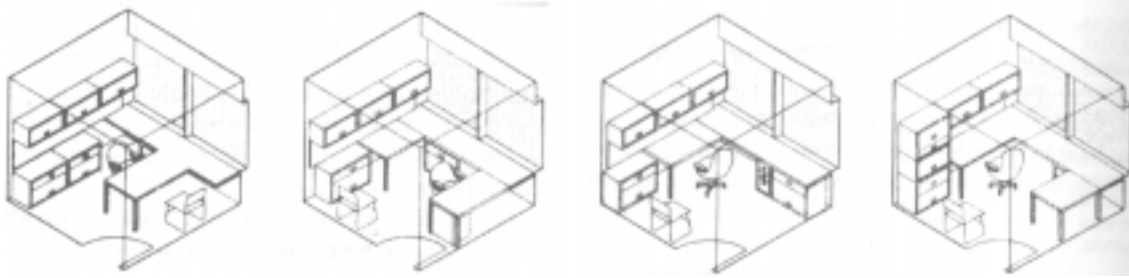
- Despachos con ventanas al exterior. La vista no tiene que ser espectacular, pero debe ofrecer la sensación de acceso al mundo exterior fuera del despacho, y debe permitir que el desarrollador pueda fijarse periódicamente en algo que este a una distancia superior a 24 pulgadas.
- Una pizarra de aproximadamente 1'2 metros cuadrados.
- Un tablón para notas de aproximadamente 1'2 metros cuadrados.
- Un acceso conveniente a los demás miembros del proyecto.
- Un acceso conveniente a una impresora de alta velocidad.
- Un acceso conveniente a una fotocopidora con alimentación automática de originales.
- Un acceso conveniente a salas de reuniones.
- Acceso conveniente a material común de oficina.

La figura I.2, de la página siguiente, muestra alguno de los diseños de oficinas productivas desarrollados para el centro de Santa Teresa de IBM. La productividad de los desarrolladores se incrementó un 11 por 100 al desplazarse a estos despachos.

I.2.2.- GESTIÓN DE LOS RIESGOS DE LOS ENTORNOS PRODUCTIVOS

Pérdida de productividad debida a mejoras de los despachos orientadas al estatus. Un posible fallo en la configuración de los entornos productivos consiste en que la directiva puede tratar la mejora en el entorno de los despachos como una mejora del estatus en lugar de una mejora de productividad.

Figura I.2. Diseños de oficinas productivas del laboratorio de Santa Teresa de IBM.



Fuente: IBM Systems Journals, vol. 17, núm. 1. Copyright 1978 International Business Machines Corporation

Como resultado, las organizaciones gastan dinero en algunos casos en mejoras estéticas que no mejoran la productividad, y en algunos ocasiones la disminuyen.

Si no se puede disponer de verdaderos despachos productivos, unas medidas más oportunas para incrementar la efectividad serían elevar la altura de los paneles de 1'5 metros a 2 metros, e incorporarles puertas. Hay que asegurarse de que las mejoras en los despachos siguen centradas en la productividad.

Tiempo de transición. El cambio de despacho implica prácticamente siempre varios problemas administrativos que tienen un efecto adverso sobre la productividad. Además, existe el inevitable tiempo de adaptación asociado con una mudanza. Y en ocasiones existe una pérdida de moral por parte de los desarrolladores que no tienen los nuevos despachos que esperaban.

El desplazamiento a despachos productivos producirá beneficios a largo plazo, pero la etapa final de un proyecto bajo presión es generalmente una mala ocasión para hacer una mudanza de este tipo. Un traslado así generalmente debe realizarse entre proyectos o en las fases iniciales de un nuevo proyecto.

Repercusiones políticas del trato preferencial a los profesionales del software. En una organización que generalmente no destina oficinas individuales a su personal, los miembros de la plantilla que no están relacionados con el software pueden interpretar la adjudicación de despachos individuales a los desarrolladores de software como un trato preferencial.

I.2.3.- EFECTOS SECUNDARIOS DE LOS ENTORNOS PRODUCTIVOS

Los desarrolladores aprecian las mejoras en sus puestos de trabajo. Las empresas que adoptan entornos productivos han encontrado que tienen una experiencia positiva sobre la satisfacción del desarrollador y las tasas de retención [Boehm, 1981].

I.2.4.- INTERACCIÓN DE LOS ENTORNOS PRODUCTIVOS CON OTROS MÉTODOS

Se puede usar los entornos productivos conjuntamente con cualquier otro método orientado a la productividad. Su efecto es independiente de los efectos de los restantes métodos.

I.2.5.- PUNTOS CRUCIALES DE LOS ENTORNOS PRODUCTIVOS

Se creía que la fuerte correlación entre el entorno de trabajo y la productividad podía ser el resultado de un factor oculto, derivado de que los mejores desarrolladores podrían tener de forma natural mejores despachos, porque han sido ascendidos. Pero no es así: desarrolladores de las mismas organizaciones con recursos similares tenían rendimientos diferentes.

Pasar de un entorno que apenas cumple mínimos a entornos productivos puede redundar en una mejora de la productividad del 100 por 100, con una productividad mínima probada superior en 2'6 veces [De Marco y Lister, 1985].

En conclusión, una organización que disponga en este momento de un entorno dentro de la media puede esperar probablemente una mejora en su productividad de desarrollo de un 20 por 100 o más situando a sus desarrolladores en entornos productivos. Para organizaciones que se encuentren por debajo o por encima de la media, la ganancia puede ser distinta según los casos.

I.2.6.- CLAVES PARA EL ÉXITO EN EL USO DE ENTORNOS PRODUCTIVOS

- Evitar mejoras de los despachos orientadas al estatus. Hay que centrarse en las características importantes de los despachos, como la intimidad, la ausencia de ruidos, las superficies de trabajo adecuadas y la capacidad de frenar las interrupciones.
- Planificar el cambio de despachos para un momento que no resulte crítico en el proyecto. Lo mejor es cambiar de despachos entre proyectos.
- Desarrollar un plan para controlar las repercusiones políticas de acomodar a los desarrolladores en despachos individuales.

1.3.- Compromiso (*Signing Up*)

Probablemente la motivación es el factor que más influye en la productividad. El compromiso es una técnica que en algunos casos ha llevado a niveles extraordinarios de motivación. Su éxito depende de tener una visión clara con la que puedan comprometerse los miembros del equipo, y en controlar activamente el proyecto para asegurarse que el equipo comprometido desarrolla el producto en una dirección aceptable.

Eficacia

Reducción potencial de la planificación nominal:	Muy buena
Mejora en la visibilidad del progreso:	Ninguna
Efecto sobre el riesgo de la planificación:	Aumenta el riesgo
Posibilidad de éxito inicial:	Media
Posibilidad de éxito a largo plazo:	Buena

Riesgos principales

- Incrementar la ineficiencia.
- Reducción de la visibilidad y control del estado.
- Disponibilidad de menos personal cualificado para el proyecto.
- Agotamiento.

Interacciones y equilibrios principales

- Sacrifica posibles disminuciones en visibilidad, control y eficiencia por un mayor incremento de la motivación.

Con la técnica del compromiso, un directivo les pide a los posibles miembros del equipo que suscriban un compromiso incondicional para que un proyecto tenga éxito. Posteriormente se permite al equipo que complete el proyecto a su aire. El compromiso produce sus beneficios para el desarrollo a partir de su increíble capacidad de motivación. Los desarrolladores que se comprometen realizan un compromiso voluntario y personal con el proyecto, y a medida llegan a alturas increíbles para hacer que el proyecto tenga éxito.

1.3.1.- USO DEL COMPROMISO

La forma en que se usa el compromiso, ofrece a los desarrolladores poco más que las recompensas intrínsecas del propio trabajo: la posibilidad de trabajar en algo importante, incrementar sus posibilidades, alcanzar una meta aparentemente imposible y llegar a algo que no haya conseguido aún nadie de la organización.

Tracy Kidder describe un equipo comprometido en el que el beneficio principal del compromiso es llamado “maquina recreativa” [Kidder, 1981]. El único beneficio obtenido es que se puede jugar de nuevo. Comprometerse en un proyecto de desarrollo, éxito, comprometerse en el siguiente proyecto interesante. Ésta es la recompensa, y es toda la recompensa que necesitan muchos desarrolladores. Parece más bien un ejercicio ilógico y masoquista. Para unos puede parecer manipulación por parte de la directiva. Para otros, representa una oportunidad esperada largo tiempo.

Definir un desafío y una visión. Hay que ofrecer una visión clara de lo que se supone que hay que realizar en el proyecto. Para hacer que la gente se comprometa, la visión tiene que ser extraordinariamente atractiva. Terminar simplemente un proyecto no basta.

Dar una oportunidad de comprometerse a las personas. La técnica del compromiso no funciona si la gente no tiene oportunidad de decidir si quiere o no comprometerse. Es posible y hay que aceptar que algunas personas no acepten el compromiso. Si ya se ha configurado el equipo, no se puede usar compromiso a menos que se este dispuesto a echar gente. El compromiso necesita implementarse al principio del proyecto o en respuesta a una crisis. No se puede iniciar a medio camino.

Una vez que los desarrolladores hayan elegido, el compromiso debe ser inequívoco. Tienen que comprometerse al éxito del proyecto sin importar como.

Compromiso a nivel de equipo. El compromiso parece funcionar mejor en equipos lo suficientemente pequeños como para tener una identidad de equipo. En grandes proyectos habría que crear pequeños equipos. Las personas necesitan identificarse con un grupo suficientemente pequeño como para que sepan que su contribución es importante, todas y cada una de ellas se sentirán responsables de terminar el producto, una vez terminado cada persona se sentirá clave, todas estarán en lo cierto.

Dar todas las atribuciones al equipo. El compromiso no funciona a menos que se deje que el equipo juegue solo. Hay que señalar la dirección, pero no especificar como llegar al final.

No usar compromiso en un proyecto con requerimientos poco firmes. Un equipo altamente motivado precisa ir rápidamente en la dirección adecuada y no dar palos de ciego. La única excepción se produce cuando el equipo en sí es responsable de la definición de los requisitos.

I.3.2.- GESTIÓN DE LOS RIESGOS DEL COMPROMISO

Incremento de la ineficiencia. Los equipos comprometidos tienden a trabajar duro en lugar de a trabajar con mayor inteligencia. Si se centran en trabajar duro, seguro que se cometerán errores que retrasarán el proyecto. Hay que vigilar el incremento de errores costosos y otras señales de que la gente no está trabajando con el cuidado necesario.

Disminución de la visibilidad del estado. En algunos casos, esta mentalidad voluntariosa hace difícil determinar el estado real del proyecto.

Pérdida de control. El equipo comprometido toma vida propia, no siempre la deseada. Forzar al equipo a cambiar la dirección da la impresión de falta de respaldo y puede resultar fatal para la moral. Controlar este riesgo requiere una valoración entre moral y eficiencia, así como entre moral y control.

Menor disponibilidad de personal para el proyecto. Desarrolladores experimentados que ya se han comprometido en proyectos anteriormente, simplemente pueden no comprometerse, si el compromiso les parece exigente, por lo que puede haber una disminución del nivel medio de experiencia en el proyecto.

Agotamiento. Lógicamente un duro trabajo requiere un duro esfuerzo.

I.3.3.- EFECTOS SECUNDARIOS DEL COMPROMISO

No tiene ningún efecto predecible y consistente sobre las características del producto. Pero al tener el equipo de desarrollo un mayor control sobre el carácter del producto, algunos atributos del mismo pueden ser mejores, peores o simplemente distintos de lo que se esperaba.

I.3.4.- INTERACCIONES DEL COMPROMISO CON OTROS MÉTODOS

Mantiene una estrecha relación con el **equipo de trabajo**. Se pueden esperar algunas **horas extra voluntarias** a cambio se les puede ofrecer un **entorno productivo**. El equipo podría resistir también técnicas más activas y laboriosas como los "Hitos Miniatura".

I.3.5.- PUNTOS CRUCIALES DEL COMPROMISO

El grado de compromiso que se obtenga variará dependiendo de la excitación que se sea capaz de generar sobre el proyecto.

I.3.6.- CLAVES PARA EL ÉXITO EN EL USO DEL COMPROMISO

- Crear una visión interesante del proyecto, para que los miembros del equipo vean algo por lo que vale la pena comprometerse.
- Hacer que el compromiso sea voluntario.
- Reforzar el equipo para que pueda tener éxito con el desafío al que se le ha motivado a responder.
- Estar preparado para controlar o aceptar las ineficacias que se deriven del hecho de que la gente trabaje duro más que de forma inteligente.

1.4.- Gestión Theory-W

En un proyecto software participan muchos implicados con intereses opuestos. Theory-W proporciona un entorno de trabajo orientado a la reconciliación de intereses. Se basa en la realización de un esfuerzo explícito para comprender lo que necesitan para “tener éxito” las distintas partes implicadas, negociar conflictos entre las condiciones de éxito de los implicados y estructurar el proyecto de forma que todos alcancen sus condiciones de éxito.

Eficacia

Reducción potencial de la planificación nominal:	Ninguna
Mejora en la visibilidad del progreso:	Muy buena
Efecto sobre el riesgo de la planificación:	Disminuye el riesgo
Posibilidad de éxito inicial:	Excelente
Posibilidad de éxito a largo plazo:	Excelente

Riesgos principales

- Ninguno

Interacciones y equilibrios principales

Diseñado para su uso combinado con el modelo de ciclo de vida en espiral.
Particularmente efectivo durante las negociaciones de planificación.
Se basa en el uso del método de negociación conveniente.

Theory-W toma su nombre de su principio más importante: *Make everyone a Winner* (Hacer que todos ganen).

La mayoría de los proyectos software comienzan con un grupo de implicados en el proyecto que tienen objetivos contrapuestos como se ve en la siguiente tabla:

Clientes	Jefes	Desarrolladores	Usuarios finales	Mantenimiento
Planificación rápida	Sin retrasos	Diseño interesante	Muchas prestaciones	Sin defectos
Presupuesto bajo	Sin sorpresas	Exploración de nuevas técnicas	Software amigable	Buena documentación
	Éxito en el proyecto	No tener trabajo desagradable	Software rápido	Fácilmente modificable
		Vida familiar	Software robusto	

Fuente: Adaptado de [Boehm y Ross, 1989]

Como se aprecia en la tabla muchos de los objetivos están en conflicto, se han remarcado con colores algunos ejemplos de objetivos en conflicto.

Theory-W defiende que hacer que todos los implicados salgan ganando en un proceso de software es una condición necesaria y suficiente para que un proyecto software se ejecute con éxito. Los conflictos entre estos grupos son la base de la mayoría de los problemas de gestión de los proyectos software, y afectan a la posibilidad de establecer los objetivos, definir las fechas de entrega, dar prioridad a las tareas y adaptarse a los cambios.

Objetivos más claros del proyecto. Theory-W comienza identificando las condiciones de “éxito” de cada uno de los implicados. La negociación que tiene lugar al comienzo del proyecto refuerza y documenta los objetivos del mismo. Establecer los objetivos de forma clara es clave para motivar a los desarrolladores, y ésta motivación es vital para un buen y rápido desarrollo.

Mejores relaciones con el cliente. Muchos proyectos implican pérdidas para una o varias de las partes implicadas, y si alguien pierde es difícil mantener unas buenas relaciones. Theory-W mejora la propia naturaleza de las relaciones porque todos están ahí para ganar.

Reducción del riesgo relacionado con el cliente. Establecer una relación Theory-W con el cliente ayuda a minimizar o eliminar riesgos como el cambio de prestaciones, comunicación lenta y gestión a pequeña escala. Ya que tiene a la vista estos riesgos relacionados con el cliente y puede obtener pronto signos de su existencia.

I.4.1.- USO DE LA GESTIÓN THEORY-W

Cuando se utiliza el método de gestión Theory-W, al comienzo del proyecto se reúne a todo el mundo y se identifican sus condiciones de éxito. Luego todos negocian para crear un conjunto de condiciones de éxito a nivel del proyecto que pueden alcanzarse razonablemente. En el resto del proyecto, la gestión consiste en controlar el proyecto asegurar que cada implicado continúe ganando. Si uno de los implicados comienza a perder, se ajusta el proyecto según corresponda.

Hacer que todos ganen. Desarrollar situaciones en las que todos los implicados ganen es una cuestión de negociación. Un proceso general que puede utilizarse para establecer una acción de ganancia a partir de ser positivos (getting to yes) es el siguiente [Fisher y Uri, 1981]:

- Separar a las personas del problema.
- Centrarse en los intereses en vez de en las posiciones.
- Inventar opciones para el beneficio mutuo.
- Insistir en el uso de criterios objetivos.

La aplicación de estos puntos a la gestión Theory-W describe los siguientes pasos:

Paso 1: Establecer las precondiciones donde todos ganen

Si no se puede imaginar cómo hacer que todos ganen, no hay que pensar siquiera en aplicar ésta gestión al proyecto. Es necesario comprender la forma en que las personas desean ganar, establecer expectativas razonables por parte de todos los implicados, hacer corresponder las tareas de las personas con sus condiciones de éxito, y establecer un entorno de apoyo.

Comprender la forma en que las personas desean ganar. Antes de comprender la forma en que desean ganar las personas, hay que asegurarse de identificar a todas las personas clave, un olvido de un implicado y el proyecto estará en un buen camino hacia el fracaso.

Hay que ponerse en el lugar de los implicados, empatizar, para comprender lo que necesitan ganar. Habitualmente el grupo más importante de entender (y el menos representado) es el de los clientes.

Establecer expectativas razonables. Si se considera que alguna de las partes tiene expectativas irrazonables, hay que reunir a los implicados para identificar expectativas y corregir errores. A menos que se puedan alinear las expectativas de los diferentes implicados, una de las partes saldrá perdiendo y el proyecto fallará.

Alguna de las siguientes acciones pueden ayudar a que las expectativas cuadren con la realidad:

- Hacer que las personas vean los temas desde los puntos de vista de los demás (Empatía).
- Relacionar las expectativas de las personas con la experiencia (hechos históricos, opinión de expertos, . . .).
- Relacionar las expectativas de las personas con modelos bien calibrados (utilizar los resultados de un modelo de estimación del software).

Adecuar las tareas de las personas con sus condiciones de éxito. Muchas veces las personas no tienen influencia sobre los aspectos del proyecto que les permitirán ganar. Esto es desmotivante, y aboca el proyecto al fracaso. En su lugar hay que asignar las tareas a las personas de forma que puedan conseguir sus propias condiciones de éxito.

El objetivo común es la necesidad de buscar situaciones creativas donde todos ganen. Algunas de estas situaciones necesitarán estructurar el proyecto de forma inusual, lo cual es perfecto, el proyecto tendrá más éxito por ello.

Proporcionar un entorno que soporte los objetivos del proyecto. El tipo de entorno dependerá del punto de vista de cada implicado.

Paso 2: Estructurar un proceso de software donde todos ganen

Es necesario establecer un plan realista del proceso, utilizar el plan para controlar el proyecto, identificar y gestionar los riesgos en los que todos pierden o unos pierden y otros ganan, y mantener comprometidas a las personas.

Establecer un plan realista. La planificación es clave para obtener éxito con Theory-W. El plan marca el camino para conseguir resultados donde todos ganen. Recoge el compromiso mutuo de los implicados en el proyecto sobre un conjunto de condiciones de beneficio mutuo y con todas las implicaciones de estar de acuerdo para conseguirlo. Las expectativas razonables hacen que el trabajo de planificación resulte más fácil, porque permiten crear un plan realista.

Utilizar el plan para controlar el proyecto. Una vez creado el plan, es necesario seguirlo. El plan proporciona un marco de trabajo para detectar déficit en las condiciones de éxito y realizar las correcciones necesarias.

Identificar y gestionar los riesgos donde todos pierden y donde unos pierden y otros ganan. La gestión de riesgos es clave en Theory-W. Para cada una de las condiciones de éxito de los implicados, hay que identificar un riesgo de no lograr esas condiciones. Luego controlar ese riesgo junto con los otros riesgos del proyecto.

Mantener comprometidas a las personas. Theory-W es un proceso orientado a las personas, y es importante mantener comprometidos a todos los implicados durante todo el proyecto. Los implicados estarán más preocupados que nadie por sus propias condiciones de éxito, y mantener comprometidos a los implicados en revisiones, planificación y otras actividades ayudará a conseguir un resultado en el que todos ganen.

Paso 3: Estructurar un producto software con el que todos ganen

Además de estructurar el proceso para tener éxito, en Theory-W hay que estructurar también el producto. Esto incluye tanto aspectos externos del mismo, qué verán los usuarios finales, como aspectos interno, qué verán desarrolladores y el personal de mantenimiento.

Tipos de proyectos que pueden usar Theory-W. La gestión Theory-W se puede comenzar a aplicar durante cualquier fase, aunque resulta más útil cuando se aplica desde el principio del proyecto. Theory-W ayuda a mantener una ejecución fluida de un proyecto con éxito, y puede ayudar a recuperar un proyecto que tiene problemas.

Theory-W funciona mejor cuando es defendido por un directivo superior o por alguien lo suficientemente alto para utilizarlo con los clientes y con los restantes implicados.

No hay límite en el tamaño o tipo de proyecto que puede utilizar Theory-W. No necesita mucho tiempo de gestión, de forma que se puede utilizar para proyectos pequeños. En los proyectos más grandes lo más importante será identificar y crear las condiciones de éxito. Theory-W es tan adecuado para el mantenimiento como para el nuevo desarrollo, y es tan adecuado para el software a medida como para el de gestión o sistemas. Los implicados específicos cambian pero no el valor de encontrar sus condiciones para tener éxito.

Función del directivo. en Theory-W además de gestionar informes directos, etcétera, también se tienen que gestionar las relaciones y objetivos de los implicados. En este esquema, el directivo es un negociador con la responsabilidad de dar soluciones al proyecto que hagan que todos ganen. También tiene que establecer los objetivos y controlar su progreso. Buscar los conflictos, enfrentarse a ellos y cambiarlos para obtener situaciones donde todos ganen.

I.4.2.- GESTIÓN DE LOS RIESGOS DE LA GESTIÓN THEORY-W

No hay riesgos particulares intrínsecos en este método. La gestión Theory-W eleva la posibilidad de que algún implicado pueda resultar perdedor a la categoría de riesgo, y eso da lugar a dos riesgos genéricos para un proyecto Theory-W.

Una de las partes comienza a entrar en una condición de pérdida. Cuando esto pasa con una de las partes, la reacción más apropiada es llevar a cabo un análisis de beneficio mutuo. Cuando se presentan problemas normalmente es posible seguir los pasos de Theory-W para crear acciones correctoras, que hagan que todos ganen o minimicen las pérdidas, comparándolas con otras acciones posibles. Las partes implicadas pueden redefinir sus condiciones de éxito conforme el proyecto va progresando.

Una condición de pérdida llega a ser inevitable. La premisa de Theory-W es que convertir a todo el mundo en ganador es una condición suficiente y *necesaria* para el éxito de un proyecto. Si se tiene la certeza de que el proyecto va a fallar, hay que pararlo tan rápido como sea posible.

I.4.3.- EFECTOS SECUNDARIOS DE LA GESTIÓN THEORY-W

Dependiendo de las condiciones de éxito de los implicados, puede proporcionar ventajas de mayor alcance que la eficiencia de la planificación, proporcionar mejoras en la facilidad de uso, mantenimiento, satisfacción del cliente y moral del equipo.

I.4.4.- INTERACCIONES DE LA GESTIÓN THEORY-W CON OTROS MÉTODOS

Theory-W es particularmente adecuada para ser utilizada con el modelo de ciclo de vida en espiral, por cada iteración de la espiral se pueden identificar las condiciones de éxito de los implicados y estas cambiarlas al comienzo de la iteración, lo que mantendrá altos niveles de satisfacción durante el proyecto. Theory-W es útil durante las negociaciones de planificación. En vez de centrarse en el estilo de negociación tradicional, se puede dejar claro que, a menos que ambas partes esperen ganar al hacer el proyecto, la parte que espera perder no tiene una base racional para seguir el juego.

I.4.5.- PUNTOS CRUCIALES DE LA GESTIÓN THEORY-W

No hay datos cuantitativos disponibles sobre el volumen de las reducciones en la planificación derivadas del uso de Theory-W, pero aparentemente reduce el riesgo de la planificación y puede mejorar la velocidad de desarrollo.

I.4.6.- CLAVES PARA EL ÉXITO EN EL USO DE LA GESTIÓN THEORY-W

- Entender cómo desean ganar las personas.
- Establecer expectativas razonables para todas las partes implicadas.
- Adecuar las tareas de las personas con sus condiciones de éxito.
- Proporcionar un entorno que soporte los objetivos del proyecto.
- Establecer un plan realista.
- Utilizar el plan para controlar el proyecto.
- Identificar y gestionar los riesgos donde todos pierden y donde unos pierden y otros ganan.
- Mantener comprometidas a las personas.
- Adecuar el producto a las condiciones de éxito de los usuarios finales y del personal de mantenimiento.

1.5.- Horas Extras Voluntarias

Las horas extras voluntarias consisten en ofrecer a los desarrolladores un trabajo significativo, de modo que junto con otras contribuciones a la motivación interna, éstos deseen trabajar más de lo que se les exige. Las horas extras ofrecen un aumento directo de la productividad, y la motivación extraordinaria da a los desarrolladores una agudeza que se transfiere a sus horas normales de trabajo. Las horas extras moderadas y voluntarias pueden utilizarse prácticamente en cualquier entorno, pero su aplicabilidad está limitada por el hecho de que las horas extras excesivas, obligatorias, se utilizan ya bastante a menudo.

Eficacia

Reducción potencial de la planificación nominal:	Buena
Mejora en la visibilidad del progreso:	Ninguna
Efecto sobre el riesgo de la planificación:	Aumenta el riesgo
Posibilidad de éxito inicial:	Media
Posibilidad de éxito a largo plazo:	Muy buena

Riesgos principales

- Penalizaciones sobre la planificación resultantes de una excesiva presión en la planificación y un exceso de horas extras.
- Reducción de la capacidad de respuesta a una necesidad de emergencia de más horas de trabajo.

Interacciones y equilibrios principales

Requiere el uso de métodos de motivación sincera y sin manipulación. Generalmente se requiere como soporte para los hitos miniatura o modelos de ciclo de vida incrementales y desarrollo en ventanas temporales (cualquier método que utilice fechas límite frecuentes).

Un exceso de horas extras y presión de planificación puede deteriorar la planificación de un desarrollo, pero unas cuantas horas extra pueden incrementar la cantidad de trabajo y mejorar la motivación.

1.5.1.- USO DE LAS HORAS EXTRAS VOLUNTARIAS

Usar la iniciativa del desarrollador, en lugar de la presión de la directiva. Incrementar la presión aplicada produce un aumento del rendimiento hasta un máximo, y luego lo lleva a cero [Weinberg, 1971].

Cuando la motivación es baja, tampoco importa la cantidad de tiempo que la gente permanece en el despacho.

La motivación de los desarrolladores con una presión media de planificación es alta. Con poco más alcanzará el máximo rendimiento, una presión adicional causa una caída del mismo (figura 1.3).

Figura I.3. Zona óptima de la presión de planificación.

Fuente: [Jones, 1991]

Resulta adecuado pedir que se trabajen algunas horas extras, pero no hay que pedir demasiadas. La clave para hacer que los desarrolladores trabajen horas extras es actuar sobre su automotivación natural.

En términos generales los cinco factores principales para motivar a los desarrolladores son: *realización, posibilidad de superación, el trabajo en sí, vida personal, oportunidad de supervisión técnica.*

Una de las formas más efectivas de motivar a los desarrolladores es infundir un sentimiento de excitación en todo el equipo de desarrollo. Sin que el equipo pierda la perspectiva de que el resultado es más importante que el trabajo duro. La importancia de centrarse en el resultado es una de las principales razones por las que las horas extras forzadas por el responsable no funcionan.

No exigir la realización de horas extras; se reducirá el rendimiento total. Esta es una de las posibles objeciones a la figura I.3. El problema relacionado con la presión para exigir más horas extras de las que desean realizar los desarrolladores consiste en que cuando la motivación comienza a caer, no solo afecta a las horas extras, también a las 40 horas *normales*. La figura I.4 muestra la relación entre la presión de planificación y las horas trabajadas, el rendimiento total y la motivación del desarrollador. El pico del rendimiento total coincide con el mismo número de horas por semana de la motivación del desarrollador.

Figura I.4. la relación entre la presión de planificación y las horas trabajadas, el rendimiento total y la motivación del desarrollador.

Como la motivación tiene mayor influencia en la productividad, al caer la motivación, el rendimiento total cae también. No cae tan bruscamente debido a que la caída en la motivación es amortiguada parcialmente por el incremento en el número de horas trabajadas.

La implicación de la figura 1.4 es clara; por encima del número de horas de trabajo empleadas voluntariamente por el trabajador, se perderá rendimiento si se presiona para que trabaje más horas extras. No importan las razones ni lo buenas que sean, a menos que los desarrolladores compartan los motivos, la motivación caerá, y el incremento de horas extras se traducirá en un menor progreso.

El número específico de horas semanales que se corresponde con el rendimiento máximo varía entre proyectos. Resulta difícil saber la cantidad de horas de trabajo que corresponden con el rendimiento óptimo de un desarrollador específico, la clave para el rendimiento máximo es buscar la motivación más alta posible, independientemente de la cantidad de horas extras que implique.

No usar horas extras para intentar recuperar el control de un proyecto. En sí mismas, las horas extras representan un signo de que un proyecto está fuera de control.

Exigir una cantidad realista de horas extras. Trabajar muchas horas puede implicar la inclusión de las tareas personales en la jornada laboral y un aprovechamiento eficaz mínimo de cada una de las horas (incluidas las horas normales).

Hay que tener cuidado con el exceso de horas extras, independientemente del motivo. El exceso de horas extras y la excesiva presión de planificación que le acompaña conducen a los siguientes problemas de planificación:

- Aumento del número de defectos.
- Aumento de las posibilidades de asumir riesgos peligrosos.
- Reducción de la creatividad.
- Incremento del desgaste.
- Incremento de la movilidad de personal.
- Disponibilidad de menos tiempo para la formación y la mejora de la organización.
- Reducción de la productividad.

Hay que ayudar a los desarrolladores a marcar su propio ritmo.

1.5.2.- GESTIÓN DE LOS RIESGOS DE LAS HORAS EXTRAS VOLUNTARIAS

Muchos de los riesgos asociados con las horas extras moderadas están relacionados con el exceso de horas extras y la presión de planificación (riesgos descritos en el punto anterior). Un riesgo adicional es:

Reducción de la capacidad de responder a necesidades de emergencia de más horas. La planificación de las horas extras se alimenta directamente de sus reservas. En un proyecto en el que todo el mundo ya está haciendo horas extras no se puede pedir más sin dañar la motivación y reducir la productividad, lo que implica que ya no se tiene margen de maniobra.

Para evitar este problema, hay que utilizar las horas extraas moderadas sólo como unaa medida correctora. No plaantar desde un principio que los desarrolladores trabajen más alla de su programación nominal. Si se tienen problemas posteriormente, se podra acudir a las reservas y poner de nuevo el proyecto en el buen camino.

I.5.3.- EFECTOS SECUNDARIOS DE LAS HORAS EXTRAS VOLUNTARIAS

El uso moderado de horas extras no tiene efectos secundarios sobre la calidad, facilidad de uso, funcionalidad, u otras características del proyecto o producto.

I.5.4.- INTERACCIONES DE LAS HORAS EXTRAS VOLUNTARIAS CON OTROS MÉTODOS

Como las horas extras realizadas por iniciativa del desarrollador son el único camino para tener éxito en el uso de horas extras, existe una relación con los métodos orientados a la **motivación**, incluyendo el **compromiso** y el **equipo de trabajo**.

Generalmente se requiere el uso de horas extras moderadas para soportar los hitos miniatura, el desarrollo en ventanas temporales, la entrega evolutiva, el prototipado evolutivo y otros métodos de desarrollo que emplean fechas límite frecuentes.

I.5.5.- PUNTOS CRUCIALES DE LAS HORAS EXTRAS VOLUNTARIAS

El punto clave de las horas extra voluntarias consiste en que un ligero incremento en las horas trabajadas al día por encima de lo estándar(de un 10 a un 20 por 100), resultará generalmente en un incremento desproporcionadamente grande en la productividad.

Desafortunadamente, en muchas organizaciones las horas extras moderadas son un método ampliamente utilizado o sustituido por horas extras excesivas y obligatorias.

Esta situación nos lleva a la interesante posibilidad de que las organizaciones en general podrían incrementar su rendimiento actual reduciendo el nivel de horas extras por debajo de las actuales.

I.5.6.- CLAVES PARA EL ÉXITO EN EL USO DE LAS HORAS EXTRAS VOLUNTARIAS

- Usar la iniciativa del desarrollador en lugar de la presión de la directiva.
- Actuar sobre las motivaciones del desarrollador.
- Pedir una cantidad de horas extraas que se puedan obtener.
- Tener cuidado con el exceso de horas extras, independientemente del motivo.

Anexo II. Contraindicaciones de estrategias alternativas

Este documento ha enfocado la gestión de recursos humanos en proyectos informáticos como una labor que tiene que surgir del sentido común, para consiguiendo una mayor motivación y huyendo de una excesiva presión, tener éxito en la consecución de los objetivos del proyecto.

Pero hay más enfoques. El más usado y conocido como método a destajo.

Este método se caracteriza por:

- Contratar al mejor personal posible.
- Buscar un compromiso total con el proyecto.
- Garantizar una autonomía casi total.
- Motivación en grado extremo.

Después se puede comprobar que se trabajan 60, 80 o incluso 100 horas a la semana hasta que bien ellos o el proyecto terminan.

Este enfoque ha generado notables éxitos, es el enfoque de desarrollo rápido más popular que se usa hoy en día. Supone la ventaja de sacar el trabajo de dos meses de un empleado con la paga de un mes. Esto puede marcar la diferencia para acabar un producto clave a tiempo de colocarlo en el mercado.

Desafortunadamente, este enfoque siempre es llevado a la práctica con más dureza que cuidado. Normalmente genera un enfoque de codificación a destajo, es una corrección rápida y tiene todos los problemas de las mismas.

A continuación se muestra una crítica a algunos aspectos de este enfoque:

El enfoque implica ganar o perder. Los factores que hacen que funcione o no son prácticamente imposibles de controlar. Este enfoque hace que sea difícil de controlar las características específicas del producto.

Provoca problemas de motivación a largo plazo. Los desarrolladores comienzan con entusiasmo y al principio emplean muchas horas extras. Pero normalmente, no es suficiente con muchas horas y fallan al intentar conseguir los compromisos del plan.

La moral se apaga y los desarrolladores se vuelven reacios a aceptar compromisos adicionales. El proyecto pierde todo aspecto de planificación o control.

No se puede repetir. Este enfoque cansa a la gente. Una empresa no puede reparar fácilmente el daño humano que provocan estos proyectos, y los estudios de estos proyectos indican que provocan un gran volumen de cambio de personal [Kidder, 1981].

Es duro para organizaciones no dedicadas al software. Aporta poca viabilidad y control a los demás implicados en el proyecto porque se basa en heroicidades individuales en vez de en la coordinación, cooperación y planificación. Lo que es bueno para una parte del proyecto no es bueno necesariamente para el proyecto completo.

Se malgastan recursos humanos exageradamente. Los desarrolladores que participan en este tipo de proyectos se olvidan de todo lo ajeno al proyecto para poder tener éxito en el mismo. Severos conflictos personales son la regla en vez de la excepción.

La siguiente tabla resume algunas de las diferencias entre el método a destajo y el compendio metodológico descrito en el presente documento:

Método a destajo	Método de este documento
Sus defensores aducen mejoras increíbles e instantáneas en el tiempo de desarrollo	Modestas mejoras instantáneas, seguidas por mejoras mayores y a más largo plazo
Necesita poca experiencia tecnológica aparte de conocimientos en codificación	Necesita una experiencia tecnológica además de conocimientos en codificación
Riesgo alto: frecuentemente falla, incluso cuando se ha hecho de la forma más efectiva posible	Riesgo bajo: pocas veces falla cuando se aplica de forma efectiva.
¿Radical?	¿Conservador?
Se emplean demasiados recursos humanos	Usa los recursos humanos de forma humana y eficiente
Ofrece poco avance en visibilidad o control. Se sabe que se ha acabado cuando se termina	Es posible personalizar el método para dar toda la visibilidad y el control necesarios
Es un método tan viejo como el software, con resultados dispares	Los elementos claves se emplean con éxito desde hace más de 15 años

Fuente: inspirado en [Davis, 1994]

Anexo III. Test Myers-Briggs

Hay una amplia batería de tests, preparados por psicólogos, que ayudan a entender por qué las personas se comportan de un modo y no de otro. Estos instrumentos permiten a los directivos tener una visión bastante clara de las raíces de los conflictos, la motivación y la productividad. Se usan para diferentes fines: contratar personal, asignar a cada persona una tarea compatible con su personalidad, determinar las destrezas de cada uno, librarse de los trabajadores que no progresan y ayudar a todos a hacer autocrítica.

Uno de estos tests, el *Thomas-Kilman Conflict Mode Instrument*, mide los compromisos de competencia, colaboración, compromiso, elusión y ajuste en situaciones de conflicto. El *T-P (Task-People) Leadership Questionnaire* examina hasta que punto un individuo en situación de trabajo se concentra en la tarea y no en las personas. El *FIRO-B Awareness Scale* examina a las personas en tres dimensiones: *inclusión* (¿Desea usted intensamente ser incluido en actividades de grupo? ¿Le gusta incluir a otros?), *control* (¿Prefiere estar en situaciones que estén bajo control de las situaciones?) y *afectividad* (¿Para usted es importante que lo quieran? ¿Expresa su afecto hacia los otros?)

Estos tests no son una panacea para resolver las dificultades organizacionales. De hecho, siempre existe el peligro de usarlos mal. Sin embargo, bien empleados, los tests pueden ser muy útiles para los gerentes de proyecto.

Posiblemente el test más útil para los directores de proyecto sea del *Myers-Briggs Type Indicator* (Indicador Tipológico de Myers-Briggs). Lo atractivo de este método es que se basa sólidamente en la teoría jungiana, ha sido sometido a serias verificaciones empíricas, es de fácil comprensión y se adapta muy bien a las situaciones propias del proyecto.

III.1.- Antecedentes del método

Las personas no se comportan todas de una sola manera, no todas tienen las mismas aspiraciones, cada una reacciona de distinto modo ante diferentes estímulos. En resumen, desde el punto de vista del comportamiento, las personas son singulares, únicas. Sin embargo, pese a su singularidad, se pueden hacer algunas generalizaciones, y a partir de ellas se llega a comprender mejor lo que motiva a la gente y la pone en acción. Algunas personas son agresivas; otras, pasivas; algunas trabajan bien en grupo; otras no pueden hacerlo; algunas son curiosas; otras, no lo son.

Carl Jung, el prestigioso psicoanalista suizo, elaboró una categorización de las personas y creó algunos tipos psicológicos. En 1923, publicó una obra en la que describía esos tipos. Su trabajo se ensambla perfectamente con las posteriores investigaciones de Katharine C. Briggs, quien tomó la teoría de Jung y la fundió con sus propias ideas. El resultado fue el *Myers-Briggs Type Indicator*, que se aplica en muchos tests psicológicos destinados a determinar el tipo psicológico.

El método Myers-Briggs categoriza a la gente según cuatro escalas, cada una de las cuales refleja una dimensión del comportamiento humano: extroversión-introversión, sensibilidad-intuición, racionalidad-afectividad y juicio-percepción. Esas cuatro escalas dan origen a dieciséis casilleros donde es posible ubicar a una persona.

Por ejemplo, alguien puede pertenecer al tipo extrovertido, intuitivo, mental y perceptivo (características del innovador); o bien al introvertido, sensato, mental, juzgador (características del administrador), y así sucesivamente.

Cada tipo psicológico posee cierto número de características bien documentadas. Si se conoce el tipo de alguien, se puede inferir rápidamente cómo se comportará en determinada circunstancia. Desde luego, esa información es útil para los gerentes de proyecto, que deben tratar constantemente con muchas personas diferentes en toda clase de circunstancias. El manejo de estas herramientas de trabajo permite llegar a entender las motivaciones de la gente y prever su conducta.

III.2.- Dimensiones

III.2.1.- LA DIMENSIÓN EXTROVERSIÓN - INTROVERSIÓN

Según el esquema de Myers-Briggs, un extrovertido es alguien orientado hacia el mundo exterior, el mundo de las personas y las cosas; y un introvertido es alguien orientado hacia el mundo interior, el de los conceptos y las ideas. Como están en sintonía con lo que sucede a su alrededor, los extrovertidos suelen ser prácticos. También les gusta hacer varias cosas al mismo tiempo. Los introvertidos, en cambio, como están dirigidos hacia adentro, son propensos a rumiar sus ideas: viven sobre todo en su mente. Por eso piensan con más profundidad que los extrovertidos.

Desde un punto de vista gerencial, cuando ponemos a un extrovertido y a un introvertido juntos en la misma sala. Entonces la situación podría ser más o menos la siguiente: al extrovertido le resultará muy difícil aguantar al introvertido porque lo considera lento, poco práctico y totalmente anacrónico. Le irrita en particular que insista en abordar un solo problema por vez y quiera buscar la solución durante una eternidad. Llega rápidamente a la conclusión de que su colega introvertido es un pelmazo. Además, piensa, no es precisamente brillante. Al mismo tiempo el introvertido tiene problemas para aguantar al extrovertido, a quien considera bastante superficial: peón de muchos oficios y maestro de ninguno; en resumen, una persona que no profundiza en nada. La tendencia del extrovertido a analizar al instante los problemas y su insistencia en saltar de un tema a otro son especialmente irritantes para el introvertido. Y, sobre todo, piensa, no es precisamente brillante.

El ejemplo muestra que dos individuos competentes, de similar inteligencia, pueden llegar a desdeñarse, simplemente, debido a su diferente orientación frente al mundo. Esta circunstancia tiene implicaciones prácticas para los gerentes de proyecto.

Consideramos a un gerente extrovertido, encargado de dirigir un proyecto de desarrollo de software de actualización. Es muy factible que el equipo técnico esté compuesto por gente introvertida que, debido precisamente a su introversión, no simpatiza con algunos aspectos de la realidad exterior, como, por ejemplo, los plazos para finalizar el trabajo. Ahora bien, el director del proyecto debería saber que la actitud despreocupada de sus colaboradores respecto de los plazos se debe a que son introvertidos y no a que deseen amargarle la vida. Debería reconocer también desde el punto de vista del equipo, él, la persona extrovertida, se interesa poco por el proyecto en sí y mucho por los aspectos superficiales, como, por ejemplo, los plazos. Una vez consciente de estas perspectivas, el director del proyecto podrá manejar a su equipo más inteligentemente que si basara su conducta en su respuesta emocional ante la aparente intransigencia de sus colaboradores.

III.2.2.- LA DIMENSIÓN SENSITIVIDAD - INTUICIÓN

Jung señaló que las personas perciben el mundo que las rodea de dos maneras. Los sensitivos atienden sobre todo a las sensaciones, utilizan de manera plena los cinco sentidos y su percepción del mundo se basa directamente en la información proporcionada por la vista, oído, tacto, gusto y el olfato. Operacionalmente se preocupan mucho por los hechos, es decir, por los datos reunidos directamente a través de los sentidos. Además, les gusta hacer uso de los sentidos.

Los intuitivos, por su parte, recogen información a través de sus sentidos y después la “procesan”. No les preocupan los hechos en sí mismos, sino las posibilidades que sugieren. Como son imaginativos, se interesan más por cómo *podrían* ser las cosas que por cómo *son* realmente. Por otra parte, casi siempre las personas imaginativas son también intuitivas.

Entre estos dos tipos también suele haber conflictos. Desde la perspectiva de los sensitivos, los intuitivos encaran las cosas de un modo superficial y voluble; y los intuitivos consideran que los sensitivos son prosaicos y carecen de imaginación.

III.2.3.- LA DIMENSIÓN RACIONAL - AFECTIVIDAD

Según Jung, las personas perciben la realidad (por medio de la sensibilidad o de la intuición) y después emiten *juicios* acerca de su significado. Algunos lo hacen a través de un proceso frío y desligado: son los racionales. Operacionalmente, estos individuos se sienten más cómodos en contacto con las cosas y los conceptos, que con la gente. Pero también hay personas que basan sus juicios en consideraciones más subjetivas, en respuestas que salen del corazón y las entrañas. En el aspecto operacional, se sienten más cómodas en contacto con la gente que con las cosas. Son los afectivos.

Un interesante rasgo de la dimensión racional-afectiva es que en la orientación de las personas hacia una tendencia o la otra hay diferencias de sexo. El 60 por 100 de los varones pertenece al tipo racional y el 60 por 100 de las mujeres, al afectivo. Valdría la pena reflexionar sobre la influencia que pueden tener en la llamada “guerra de sexos” los procesos fundamentalmente diferentes a través de los cuales los hombres y las mujeres analizan el mundo que los rodea.

III.2.4.- LA DIMENSIÓN JUICIO - PERCEPCIÓN

En el esquema de Myers-Briggs, la cuarta dimensión examina hasta qué punto las personas se sienten obligadas a sacar conclusiones sobre el mundo circundante. Algunos individuos formulan un juicio inmediatamente, opinan sobre todo y sobre todos. Los cabos sueltos los ponen nerviosos. Siempre toman una decisión con rapidez; jamás la postergan. Operacionalmente, se sienten cómodos dentro del orden y la planificación. El aspecto negativo es que suelen ser rígidos y estrechos de miras y que corren el riesgo de emitir juicios prematuramente. Son los juzgadores.

En cuanto al otro tipo, está compuesto por gente que prefiere diferir la emisión de un juicio hasta contar con más información. Estas personas son flexibles y abiertas. Lamentablemente corren el riesgo de ser desorganizadas y de caer en la trampa de la postergación, porque siempre están esperando recabar más datos. Son los preceptivos.

Los conflictos que tienen lugar entre preceptivos y jugadores son obvios. La obra de teatro *Una extraña pareja*, de Neil Simon, presenta precisamente las diferencias de puntos de vista de dos hombres que viven en la misma casa: uno es organizado y fastidioso; el otro, libre y desorganizado. Esta obra es un estudio del caso y muestra los problemas que pueden surgir cuando un individuo del tipo juzgador y otro del tipo perceptivo tienen que mantener una relación estrecha.

III.3.- Aplicar a los proyectos la teoría de los tipos psicológicos

Las personas a las que se administra el test Myers-Briggs suelen sorprenderse por la extraordinaria exactitud con que este instrumento describe sus características psicológicas.

Sin embargo, no hay magia en esto. El test consiste en una serie de preguntas que identifican las preferencias de la persona. Si es docente, le preguntarán, por ejemplo, si prefiere transmitir información concreta o exponer teorías. Si prefiere transmitir información fáctica, ello indica una preferencia por la sensibilidad sobre la intuición. Al responder una batería de preguntas, se revelan las preferencias generales de la persona sobre cada una de las dieciséis categorías. Las personas que caen dentro de determinada categoría comparten gran número de características psicológicas. Así, si se sabe el tipo de una persona en la escala de Myers-Briggs, se pueden describir con exactitud algunos aspectos fundamentales de su personalidad, aun sin conocerla personalmente.

Conocer el método Myers-Briggs constituye una gran ayuda para los directores de proyecto, quienes podrán así manejar mejor a la gente en varias áreas importantes: selección de personal, diagnóstico de las raíces de los conflictos, mejoramiento de las relaciones con el equipo, autoconocimiento de los directivos.

Anexo IV. Prevención de riesgos laborales.