

25

Gestión de personal

Objetivos

El objetivo de este capítulo es mostrar la importancia de las personas en el proceso de ingeniería del software. Cuando termine de leer este capítulo:

- comprenderá algunos de los puntos involucrados en la selección y retención de personal en una organización de desarrollo de software;
- comprenderá los factores que influyen en la motivación individual y sus implicaciones para los gestores de proyectos software;
- comprenderá los elementos clave del trabajo en equipo, principalmente la composición, la cohesión, la comunicación y la organización;
- habrá sido introducido en el Modelo de Madurez de la Capacidad del Personal —un modelo que es un marco de trabajo para resaltar las capacidades de los ingenieros del software en una organización.

Contenidos

25.1 Selección de personal

25.2 Motivación

25.3 Gestionando grupos

25.4 El Modelo de Madurez de la Capacidad del Personal

El personal que trabaja en una organización software es su principal activo. Representa el capital intelectual, y es misión de los gestores asegurar que la organización obtenga los mejores beneficios posibles al invertir en las personas. En las compañías y economías con éxito, esto se cumple cuando las personas son respetadas por la organización. Dichas personas deberán tener un nivel de responsabilidad y se les deberán asignar premios de acuerdo con sus capacidades.

Por lo tanto, la gestión efectiva se refiere a la gestión del personal en una organización. Los gestores de proyectos tienen que resolver problemas técnicos y no técnicos, utilizando a las personas de su equipo de la forma más efectiva posible. Tienen que motivar a la gente, planificar y organizar su trabajo y asegurar que éste se realice de manera adecuada. La mala gestión del personal es uno de los factores más importantes en el fracaso de los proyectos.

Desgraciadamente, esta mala gestión es demasiado común en la industria del software. Los gestores fallan al tomar nota de las limitaciones individuales e imponer metas inalcanzables a los equipos del proyecto. Equiparan la gestión con reuniones, donde la gente colabora en el proyecto. Pueden aceptar nuevos requerimientos sin el análisis adecuado de lo que esto significa para el equipo del proyecto. A veces, ven su papel como una explotación de su personal, en lugar de trabajar con ellos para que su trabajo pueda contribuir a metas tanto organizacionales como personales.

Desde nuestro punto de vista, hay cuatro factores críticos en la gestión de personal:

1. *Objetividad.* El personal del proyecto debe ser valorado de forma equitativa. Mientras que alguien no espere, que todas las recompensas sean idénticas, la gente sentirá que su contribución en la organización es infravalorada.
2. *Respeto.* Las personas tienen diferentes habilidades y los gestores deben respetar estas diferencias. Todos los miembros del equipo deben dar una oportunidad para hacer una contribución. En algunos casos, por supuesto, usted se encontrará con gente que simplemente no se ajusta a un equipo, y no puede continuar, pero es importante no sacar conclusiones acerca de esto.
3. *Incorporación.* La gente contribuye efectivamente cuando siente que se la escucha y se toma nota de sus propuestas. Es importante desarrollar un entorno de trabajo donde todos los puntos de vista, hasta el de la persona más novata, se tomen en consideración.
4. *Honestidad.* El gestor de proyectos siempre debe de ser honesto con lo que está yendo bien y con lo que está yendo mal en el equipo. Debe ser honesto en lo que se refiere a su nivel de conocimientos técnicos y debe dar la razón al personal con más conocimientos cuando sea necesario. Si no es honesto, se encontrará fuera de lugar en algunas ocasiones, y perderá el respeto del grupo.

El objetivo en este capítulo es introducir al lector en algunos elementos que los gestores de proyectos software pueden encontrar, y proveerle de información que le ayude a entender estos elementos. La gestión, desde mi punto de vista, sólo puede ser aprendida con la experiencia; por lo tanto, el papel del libro es ayudarle a aprender de experiencias previas. Para llegar a ser un buen gestor de personal no será suficiente con la simple lectura de este capítulo. Sin embargo, es de esperar que este material ayude a entender los problemas que los gestores encuentran cuando dirigen equipos de individuos con talento técnico.

25.1 Selección de personal

Una de las tareas más importantes de un gestor de proyectos es seleccionar al personal que trabajará en el proyecto. En casos excepcionales, los administradores de proyectos pueden de-

signar a las personas que mejor se adecuan a un trabajo, independientemente de sus otras responsabilidades o de las consideraciones de presupuesto. Sin embargo, por lo general, los administradores de proyectos no tienen una libre elección de personal. Tienen que utilizar a quien esté disponible en una organización, tienen que encontrar a las personas muy rápidamente y cuentan con un presupuesto limitado. Esta limitación de presupuesto restringe el número de costosos ingenieros con experiencia que pueden trabajar en el proyecto.

La decisión de quién será designado para un proyecto, por lo general, se lleva a cabo utilizando tres tipos de información.

1. La información suministrada por los candidatos acerca de sus conocimientos y experiencia (su CV). Ésta suele ser la información más fiable de la que se dispone para juzgar qué candidatos son adecuados.
2. Información obtenida al entrevistar a los candidatos. Las entrevistas pueden darnos una buena visión de qué candidatos son buenos comunicadores y cuándo éstos tienen buenas habilidades sociales. Sin embargo, las pruebas han mostrado que los entrevistadores son obligados a aceptar o rechazar candidatos haciendo rápidos juicios subjetivos. Por lo tanto, las entrevistas no son métodos fiables para juzgar las capacidades técnicas.
3. Recomendaciones de otras personas que han trabajado con los candidatos. Esto puede ser objetivo cuando se conoce a la persona que hace la recomendación. De otro modo, las recomendaciones pueden no ser ciertas, por lo que deben tener poco peso en la decisión a la hora de formar el equipo.

La Figura 25.1 muestra un pequeño caso de estudio de los elementos que podemos encontrar cuando evaluamos personal. Algunas lecciones típicas que muestra el caso de uso son:

1. Si usted está buscando personas con determinadas habilidades dentro de la compañía, el gestor del proyecto en los cuales trabajan estas personas, puede no querer perderlos. A veces, tendremos que aceptar que personas trabajen a tiempo parcial en nuestro proyecto.
2. Pocos candidatos con determinadas habilidades, como diseño de interfaz de usuario e interfaz hardware. Usted puede no tener un amplio abanico de candidatos para estas tareas, especialmente si la compañía no esta cerca de otras industrias software.
3. Los recién titulados pueden no tener las habilidades que usted necesita, pero son normalmente entusiastas y pueden haber tenido contacto con la tecnología más actual.
4. No intente siempre contratar a la persona con más conocimientos técnicos para un trabajo de desarrollo técnico. En el caso de estudio, pudo ser necesaria la interacción con los usuarios ancianos y Alice decidió que Carol sería más comprensiva con sus problemas.

En la Figura 25.2 se muestra una tabla de los factores que pueden influir en su decisión cuando esté seleccionando personal. Los factores más importantes varían dependiendo del dominio de la aplicación, del tipo de proyecto y de las habilidades de los otros miembros del equipo del proyecto.

El gestor de proyectos puede encontrar problemas para descubrir gente con habilidades y experiencia apropiadas. Puede tener que formar un equipo utilizando ingenieros inexperimentados. Esto puede acarrear problemas porque los miembros del equipo no comprenden el dominio de la aplicación o la tecnología. Sin embargo, en proyectos de larga duración, entender los conceptos y el dominio de la aplicación es más importante que los conocimientos específicos, en particular los lenguajes y métodos de programación. Por lo tanto, a menos que

Caso de estudio 1. Eligiendo a los miembros del equipo

Alice es una gestora de proyectos software que trabaja en una compañía que desarrolla sistemas de alarma. Esta compañía desea entrar en el creciente mercado de tecnología de asistencia para ayudar a ancianos y discapacitados a vivir independientemente. A Alice se le encargó formar un grupo de seis desarrolladores para implementar productos basados en la tecnología de alarmas de la compañía. Su primera misión era seleccionar a los miembros del equipo, los cuales podían ya estar dentro de la compañía o fuera de ella.

Para seleccionar al equipo, Alice revisó en primer lugar las habilidades que se iban a necesitar:

- Experiencia con la tecnología de alarmas existente, y su reutilización.
- Experiencia en el diseño de interfaces, ya que los usuarios no van a tener conocimientos y pueden ser discapacitados; de ahí que necesiten facilidades como fuentes de tamaño variable, etc.
- Idealmente, alguien que tenga experiencia en diseño de sistemas basados en tecnología de ayuda a discapacitados. Por otro lado, alguien con experiencia en interfaces con unidades hardware ya que todos los sistemas que se comenzaban a desarrollar incluían algo de control de hardware.
- Habilidades generales de desarrollo.

La siguiente etapa fue encontrar gente en la compañía con las habilidades necesarias. A pesar de que la compañía se había extendido significativamente y tenía poco personal disponible, lo mejor que Alice pudo negociar fue la ayuda de un experto en alarmas (Fred) durante dos días a la semana. Por lo tanto, decidió poner un anuncio para captar el personal para el nuevo proyecto, donde aparecían los atributos que ella deseaba:

- Experiencia en programación en C. Ella había decidido desarrollar todo el software en C.
- Experiencia en diseño de interfaz de usuario. Un diseñador de interfaz de usuario es esencial, pero podría no ser necesario a tiempo completo.
- Experiencia en interfaces hardware con lenguaje C y utilizando sistemas de desarrollo remoto. Todos los dispositivos utilizados tienen interfaces complejos.
- Personalidad comprensiva ya que tenía que relacionarse y trabajar con personas ancianas las cuales iban a proveer los requerimientos y a probar el sistema.

Alice obtuvo 30 respuestas al anuncio, y de los aspirantes, fue posible identificar candidatos adecuados con conocimientos de interfaz con hardware (Dorothy) y experiencia con interfaces de usuario (Ed). Decidió contratar a dos recién titulados (Brian y Bob) que tenían experiencia en programación en C, pero que tendrían que ser formados en la compañía.

Lo único que quedaba era designar a un programador senior para unirse al equipo de desarrollo y Alice tenía dos opciones. Carol contaba con varios años de experiencia en C y recientemente había tenido una parada en su carrera para tener un niño. Dave poseía una experiencia similar en programación y era un entusiasta de la programación. Gastaba mucho de su tiempo trabajando en proyectos open source y tenía un conocimiento teórico de C y C++.

Después de entrevistar a ambos, Alice decidió ofrecer el trabajo a Carol a pesar de que Dave tenía profundos conocimientos de programación.

Figura 25.1
Selección
de personal.

se necesitan conocimientos específicos en un lenguaje de programación para un proyecto de corta duración, es mejor elegir a personal con capacidad para resolver problemas o con dominio de la aplicación. Es relativamente sencillo aprender un nuevo lenguaje, pero mucho más difícil desarrollar el conocimiento conceptual necesario para resolver problemas complejos.

Algunas compañías prueban a los candidatos. Incluyen pruebas de aptitud a la programación y pruebas psicotécnicas donde los candidatos completan una serie de ejercicios en un periodo de tiempo relativamente breve. Las pruebas psicotécnicas están dirigidas a conseguir el perfil psicológico del candidato, indicando sus capacidades y habilidades para ciertos tipos de tareas. Algunos gestores consideran que estas pruebas son inútiles; otros piensan que proveen información que ayuda a la selección del personal. Se duda si las pruebas de aptitud pro-

Experiencia en el dominio de la aplicación	Para desarrollar bien un sistema, los desarrolladores deben entender el dominio de la aplicación. Es esencial que algunos miembros del equipo de desarrollo tengan alguna experiencia en el dominio de la aplicación.
Experiencia en la plataforma	Este factor es significativo si la programación a bajo nivel es necesaria. En otro caso, no es un atributo crítico.
Experiencia en el lenguaje de programación	Normalmente esto sólo es importante para proyectos de corta duración donde no existe tiempo suficiente para aprender un nuevo lenguaje. Mientras que aprender el lenguaje propiamente dicho no es difícil, empezar a utilizar las librerías y componentes de forma competente puede llevar varios meses.
Habilidad para resolver problemas	Esto es muy importante para ingenieros de software, los cuales tienen que resolver constantemente problemas técnicos. Sin embargo, es casi imposible de juzgar sin conocer el trabajo del candidato.
Soporte educativo	Esto provee un indicador de los fundamentos básicos que el candidato debe conocer y de la habilidad para aprender. Este factor cada vez es más irrelevante, puesto que los ingenieros obtienen experiencia a través de los proyectos.
Habilidad de comunicación	Esto es importante debido a que el personal del proyecto necesita comunicarse oralmente y por escrito con los otros ingenieros, administradores y clientes.
Adaptabilidad	La adaptabilidad se valora observando las diversas experiencias obtenidas por los candidatos. Éste es un atributo importante puesto que indica una habilidad para aprender.
Actitud	El personal del proyecto debe tener una actitud positiva con respecto a su trabajo y debe estar deseoso de aprender nuevas habilidades. Éste es un atributo importante, pero a menudo muy difícil de valorar.
Personalidad	Éste es un atributo importante pero difícil de valorar. Los candidatos deben ser razonablemente compatibles con los otros miembros del equipo. Ningún tipo de personalidad es más o menos adecuada para la ingeniería de software.

Figura 25.2 Factores que determinan la selección de personal.

veen información útil acerca de la capacidad para resolver problemas. La resolución software de problemas complejos es un proceso iterativo de la larga duración. No parece que las habilidades necesarias para la resolución de problemas complejos sean comparables a las habilidades necesarias para completar pruebas de aptitud simples.

La falta de personal técnico con experiencia puede ser resultado de que, en algunas organizaciones, las personas con habilidades técnicas rápidamente alcanzan una meta en sus carreras. Para progresar, deben tomar responsabilidades administrativas. La promoción de estas personas a un status administrativo significa perder el ejercicio de sus habilidades técnicas. Para evitar esta pérdida, algunas compañías desarrollaron estructuras paralelas de carreras técnicas y administrativas de igual valor. Las personas con experiencia técnica se consideran al mismo nivel que los administradores. Si un ingeniero desarrolla su carrera, se puede especializar en actividades técnicas o de gestión y moverse entre ambas sin la pérdida de su status o salario.

25.2 Motivación

Uno de los papeles de los gestores de proyectos es motivar a las personas que trabajan con ellos. Motivación significa organizar el trabajo y su entorno para estimular a las personas a trabajar de la forma más efectiva posible. Si las personas no son motivadas, no se interesarán

en el trabajo que hacen. Éstas trabajarán lentamente, cometerán más errores y no contribuirán a las metas del equipo ni de la organización.

Maslow (Maslow, 1954) sugiere que la gente sea motivada satisfaciendo sus necesidades y que estas necesidades están organizadas en una serie de niveles, que se muestran en la Figura 25.3. El nivel más bajo de esta jerarquía representa las necesidades fundamentales como comer, dormir, etc., y a continuación, en el siguiente nivel, la necesidad de sentirse seguros en un entorno. Las necesidades sociales se refieren a la necesidad de sentirse parte de un grupo social. Las necesidades de autoestima se refieren a sentirse respetados por los otros, y las necesidades de autorrealización se refieren a desarrollo personal. Las prioridades humanas son satisfacer las necesidades de los niveles inferiores, como estar hambriento, antes de satisfacer las necesidades, más abstractas, de niveles superiores.

Las personas que trabajan en organizaciones de desarrollo de software no están generalmente hambrientas ni sedientas y por lo general no se sienten físicamente amenazadas por su entorno. Por lo tanto, asegurar la satisfacción de las necesidades sociales, de estima y de autorrealización es más importante desde un punto de vista administrativo:

1. Satisfacer las necesidades sociales significa permitir que la gente tenga tiempo para conocer a sus compañeros de trabajo y proveer lugares para que se conozcan. Esto es relativamente fácil cuando todos los miembros del grupo de desarrollo trabajan en el mismo lugar, pero cada vez es más habitual que los miembros del equipo no estén localizados en el mismo edificio o incluso en la misma ciudad o estado. Pueden trabajar en diferentes organizaciones o desde casa la mayor parte del tiempo. Las comunicaciones electrónicas, como el e-mail y la videoconferencia, pueden ser usadas como ayuda al teletrabajo. Sin embargo, la experiencia con comunicaciones electrónicas demuestra que éstas no satisfacen realmente las necesidades sociales. Si el equipo de trabajo está distribuido, deberían organizarse encuentros periódicos cara a cara, para que la gente haga un intercambio de experiencias directamente con los otros miembros del equipo. A través de esta interacción, la gente llega a formar parte de un grupo social y puede ser motivada por las metas y prioridades del grupo.
2. Para satisfacer las necesidades de estima, se necesita mostrar a la gente que es de gran valor para la organización. El reconocimiento público de los logros es una forma sencilla pero efectiva de hacer esto. Obviamente, las personas también deben sentir que se les paga de acuerdo con el nivel que reflejan sus habilidades y experiencia.
3. Finalmente, para satisfacer las necesidades de autorrealización se necesita dar a las personas responsabilidad de su propio trabajo, asignarles tareas cada vez más exigentes (pero no imposibles) y proveerles un programa de capacitación donde puedan desarrollar sus habilidades.



Figura 25.3
Jerarquía
de las necesidades
humanas.

En la Figura 25.4 se ilustra un problema de motivación con el que se encuentran los gestores frecuentemente. En este ejemplo, un miembro competente del grupo pierde interés en el trabajo y en el grupo. La calidad de su trabajo decae y empieza a ser inaceptable. Esta situación tiene que ser resuelta con rapidez. Si no se zanja el problema, los otros miembros del grupo llegarán a estar descontentos y a sentir que el reparto del trabajo es injusto.

En este ejemplo, Alice trata de descubrir si las circunstancias personales de Dorothy podrían ser el problema. Los problemas personales normalmente afectan a la motivación, ya que la gente no puede concentrarse en su trabajo. Se debe dar tiempo y ayudar a la gente a resolver estas situaciones, aunque también debe dejarse claro que los empleados tienen unas responsabilidades que cumplir.

De hecho, el problema de motivación de Dorothy es uno de los que surgen frecuentemente cuando el proyecto se desarrolla en una dirección imprevista. La gente que espera hacer un tipo de trabajo puede hacer algo totalmente diferente. Esto comienza a ser un problema cuando las personas quieren desarrollar unas habilidades en alguna dirección que es diferente a la que toma el proyecto. En esas circunstancias, se puede decidir que miembros deberían dejar el equipo y encontrar oportunidades en otro lugar. En esta situación, Alice decide convencer a Dorothy diciéndole que ampliar su experiencia es un paso positivo en su carrera. Ella da a Dorothy más autonomía de diseño y organiza cursos de ingeniería de software que le proporcionen más oportunidades cuando el proyecto haya finalizado.

El principal problema que presenta el modelo de Maslow con la motivación es que toma exclusivamente el punto de vista del individuo. No tiene en cuenta que de hecho la gente se siente parte de una organización, un grupo profesional, y por lo general de una cultura. No se trata simplemente de satisfacer las necesidades sociales; las personas pueden ser motivadas a través de ayudar al grupo a conseguir las metas comunes.

Ser miembro de un grupo cohesivo es altamente motivador para la mayoría de la gente. A las personas con trabajos que no les hacen sentirse realizadas, frecuentemente les gusta ir al trabajo porque están motivadas por personas que trabajan con ellas y por el trabajo que esas personas hacen. Por lo tanto, es tan bueno pensar sobre la motivación individual, como pen-

Caso de estudio 2: Motivación

El proyecto de tecnología de asistencia de Alice empieza bien. Existen buenas relaciones entre los miembros del equipo, y se desarrollan ideas nuevas y creativas. La compañía decide desarrollar un sistema de mensajes peer-to-peer usando televisiones digitales enlazadas a la red de alarmas. Sin embargo, unos meses más tarde, Alice nota que Dorothy, la experta en diseño hardware, empieza a llegar al trabajo tarde, la calidad de su trabajo se deteriora, y cada vez más parece que no se está comunicando con los otros miembros del equipo.

Alice habla acerca del problema informalmente con otros miembros del equipo para tratar de descubrir si las circunstancias personales de Dorothy han cambiado y si es posible que esto la esté afectando en el trabajo. Ellos no saben nada, y Alice decide hablar con Dorothy para tratar de comprender su problema.

Después de negar inicialmente que hay un problema, Dorothy admite que ha perdido interés en su trabajo. Ella esperaba poder utilizar y desarrollar sus habilidades en interfaces hardware. Sin embargo, la dirección que había tomado el producto le dejaba pocas oportunidades para esto. Básicamente, ella trabajaba como programadora en C con otros miembros del equipo. Mientras admite que el trabajo está cambiando, la preocupa que no está desarrollando sus habilidades en interfaces hardware. Está preocupada, porque encontrar un trabajo donde se trabaje con interfaces hardware será difícil después de este proyecto. Como no quiere disgustarse con el equipo revelando que está pensando en el siguiente proyecto, ha decidido que lo mejor es reducir al mínimo las conversaciones con ellos.

Figura 25.4
Motivación
del individuo.

sar en cómo el grupo puede ser motivado para alcanzar las metas de la organización. En la sección siguiente se expondrá la gestión de grupos.

En un estudio psicológico de motivación, Bass y Dunteman (Bass y Dunteman, 1963) clasificaron a los profesionales en tres tipos:

1. *Los orientados a tareas*, quienes están motivados por el trabajo que hacen. En la ingeniería del software, son técnicos que están motivados por el reto intelectual de desarrollo software.
2. *Los orientados a sí mismos*, quienes están principalmente motivados por los éxitos y el reconocimiento personal. Están interesados en el desarrollo de software como medio para hacer cumplir sus propios propósitos.
3. *Los orientados a la interacción*, quienes están motivados por la presencia y acciones de sus compañeros de trabajo. Puesto que el desarrollo de software está cada vez más centrado en el usuario, los individuos orientados a la interacción están cada vez más involucrados en ingeniería de software.

Las personalidades orientadas a la interacción por lo general gustan de trabajar como parte de un grupo, mientras que las orientadas a las tareas y las orientadas a sí mismas a menudo prefieren trabajar solas. Es más probable que las mujeres estén más orientadas a la interacción que los hombres. A menudo, ellas son comunicadoras más efectivas. Se hablará de la mezcla de estas personalidades dentro de grupos en la Sección 25.3.2.

Cada motivación individual está compuesta de elementos de cada clase, pero un tipo de motivación es por lo general dominante en un momento dado. Sin embargo, las personalidades no son estáticas y los individuos pueden cambiar. Por ejemplo, el personal técnico que siente que no se le está remunerando debidamente se puede convertir en orientado a sí mismo y anteponer los intereses personales a las cuestiones técnicas.

25.3 Gestionando grupos

Mucho del software profesional es desarrollado por equipos de proyectos que varían en tamaño desde dos hasta varios cientos de personas. Sin embargo, puesto que es claramente imposible que cada persona en un equipo grande trabaje con todos al mismo tiempo y de forma eficaz en un solo problema, estos grandes equipos por lo general se dividen en varios grupos. Cada grupo es responsable de un subproyecto que desarrolla algún subsistema. Como regla general, los grupos de proyectos de ingeniería de software normalmente no tienen más de ocho o diez miembros. Cuando se utilizan grupos pequeños, los problemas de comunicación se reducen. El grupo completo puede reunirse alrededor de una mesa y llevar a cabo reuniones en una oficina.

Por lo tanto, hacer que un grupo trabaje eficazmente es una tarea crítica de gestión. Obviamente, es importante que el grupo tenga el equilibrio correcto de habilidad y experiencia técnicas y de personalidades. Sin embargo, los grupos exitosos son más que una simple colección de individuos con el equilibrio correcto de habilidades. Un buen grupo tiene un espíritu de equipo de tal forma que las personas involucradas se motivan por el éxito del grupo, así como por sus propias metas personales.

Existen varios factores que influyen en el trabajo en grupo:

1. *La composición del grupo*. ¿Existe el equilibrio correcto de habilidades, experiencia y personalidades en el equipo?

2. *La cohesión del grupo.* ¿Piensa el grupo en sí mismo como un equipo más que como una colección de individuos que trabajan juntos?
3. *La comunicación del grupo.* ¿Se comunican los miembros del grupo de forma efectiva uno con otro?
4. *La organización del grupo.* ¿Está organizado el equipo de tal forma que cada uno se siente valorado y satisfecho con su papel en el grupo?

25.3.1 La composición del grupo

Como se comentó en la Sección 25.2, muchos ingenieros de software están motivados fundamentalmente por su trabajo. Los grupos de desarrollo software, por tanto, están frecuentemente compuestos por personas que tienen sus ideas acerca de cómo deben resolverse los problemas técnicos. Éste es un punto a partir del cual aparecen regularmente problemas: los estándares de interfaz se pasan por alto, los sistemas comienzan a rediseñarse según han sido codificados, embellecimientos innecesarios del programa y otros más. A ser posible, se debe tratar de seleccionar un grupo de personas donde se eviten estos problemas.

Un grupo que tiene personalidades complementarias puede trabajar mejor que un grupo seleccionado exclusivamente por sus habilidades técnicas. La gente motivada por su trabajo es la más fuerte técnicamente. Las personas orientadas a sí mismas serán probablemente las mejores para llevar adelante el trabajo hasta finalizar las tareas. La gente orientada a la interacción facilitará las comunicaciones con el grupo. Es particularmente importante tener gente orientada a la comunicación en un grupo. A ellos les gusta hablar con la gente y pueden detectar tensiones y desacuerdos en etapas tempranas, por lo que tienen un gran impacto sobre el grupo.

En el caso de estudio expuesto en la Figura 25.5, se muestra cómo Alice, la gestora del proyecto, ha tratado de crear un grupo con personalidades complementarias. Este grupo en particular tiene una buena mezcla de personas orientadas a la interacción y a las tareas, pero ya se ha indicado en la Figura 25.4 de cómo la personalidad orientada a sí misma de Dorothy puede causar problemas. El papel de Fred con dedicación parcial como experto del dominio puede ser también un problema aquí. El está muy interesado en mejoras técnicas, y puede no interactuar correctamente con otros miembros del grupo. El hecho de que él no sea siempre parte del equipo, implica que él no se identifica bien con las metas del equipo.

Caso de estudio 3: Composición del grupo

En la creación de un grupo para el desarrollo de un sistema de tecnología de asistencia, Alice es consciente de la importancia de la selección de miembros con personalidades complementarias. Mientras estaba entrevistando a la gente, trató de evaluar si los entrevistados orientados a la tarea, orientados a sí mismos u orientados a la interacción. Sintió que al principio ella era del tipo orientado a sí mismo, y que el proyecto le permitiría ser reconocida como gestor senior y ser ascendida. Por lo tanto, buscó a una o dos personas orientadas a la interacción y quería individuos orientados a la tarea para completar el equipo. La evaluación final a la que llegó fue la siguiente:

Alice – orientada a sí misma
 Brian – orientado a la tarea
 Bob – orientado a la tarea
 Carol – orientada a la interacción
 Dorothy – orientada a sí misma
 Ed – orientado a la interacción
 Fred – orientado a la tarea

Figura 25.5
Composición
del grupo.

En algunas ocasiones es imposible elegir un grupo con personalidades complementarias. En este caso de estudio, el gestor de proyecto tiene el control del grupo, por lo que las metas individuales no trascienden a los objetivos organizacionales y del grupo. Este control es fácil de lograr si todos los miembros participan en todas las etapas del proyecto. Las iniciativas individuales son más comunes cuando los miembros del grupo están recibiendo instrucciones, sin ser conscientes de la parte que su tarea juega dentro del proyecto.

Por ejemplo, supongamos que a un ingeniero se le da un diseño de un programa para codificar y observa posibles mejoras de diseño. Si estas mejoras se implementan sin comprender la razón del diseño original, podrían tener implicaciones adversas sobre otras partes del sistema. Si todos los miembros del grupo están inmersos en el diseño desde el principio, entenderán las razones de las decisiones de diseño. Se identificarán con estas decisiones en lugar de oponerse a ellas.

Un papel importante es el de líder del grupo. Éste puede ser el responsable de proveer la dirección técnica y gestión del proyecto. Los líderes del grupo deben conservar el trabajo rutinario del grupo, asegurándose de que la gente está trabajando eficazmente y de la forma indicada por el gestor del proyecto en el planning.

Los líderes son normalmente designados por el gestor general del proyecto. Sin embargo, el líder designado puede no ser un verdadero líder en el grupo, siendo más preocupante cuanto más se aleje del liderazgo técnico. Los miembros del grupo pueden ver a otro miembro como líder. Puede tratarse de un ingeniero más competente técnicamente, o puede ser un motivador mejor que el líder del grupo designado.

A veces, resulta eficaz separar al líder técnico de la administración del proyecto. Las personas que son competentes a nivel técnico no siempre son los mejores administradores. El darle un rol administrativo puede reducir su valía dentro del grupo. Es mejor ayudarlo con un administrador que lo libere de las tareas rutinarias.

Imponer a un líder que el grupo no quiera, normalmente causará tensiones. Los miembros del equipo no respetarán al líder y pueden perder la lealtad al grupo a favor de metas individuales. Éste es un problema concreto de las áreas que cambian rápidamente, como es la ingeniería del software, donde los nuevos miembros pueden estar más actualizados y mejor preparados que los líderes experimentados del grupo. Algunas personas con experiencia pueden sentirse mal por la imposición de un líder joven con nuevas ideas.

25.3.2 Cohesión

En un grupo cohesivo, los miembros piensan que es más importante el grupo que los individuos. Los miembros de un grupo cohesivo son leales al grupo. Se identifican con las metas del grupo y con los otros miembros del grupo. Intentan proteger al grupo, como entidad, de interferencias externas. Esto hace que el grupo sea fuerte y que sea capaz de sobrellevar problemas y situaciones inesperadas. El grupo hará frente a los problemas mediante la ayuda y soporte mutuo.

Las ventajas de un grupo cohesivo son las siguientes:

- 1: *Puede crearse un grupo que utilice estándar de calidad.* Como este estándar se establece por consenso, probablemente se observará mejor que si fuese externo e impuesto al grupo.
2. *Los miembros de un grupo trabajan juntos.* Las personas del grupo pueden aprender unas de otras. Las inhibiciones causadas por la ignorancia son minimizadas, así como el aprendizaje mutuo los anima.

3. *Se puede practicar la programación sin ego puede ser practicada.* Los programas son reconocidos como propiedad del grupo, no de los individuos.

Programación sin ego (Weinberg, 1971) es un estilo de grupo de trabajo en el que los diseños, programas y otros documentos son considerados como una propiedad común del grupo en lugar de el individuo que los ha escrito. En la cultura de programación sin ego, es más probable que las personas ofrezcan su trabajo para inspección por otros miembros, para aceptar críticas y para trabajar con el grupo a fin de mejorar el programa. Los grupos cohesivos son mejores porque todos los miembros sienten que comparten la responsabilidad del software. La idea de la programación sin ego es fundamental en la programación extrema (Beck, 2000), analizada en el Capítulo 17. En la programación extrema, uno de los principios básicos es la mejora constante del código del sistema, independientemente de quién escribió el código.

Además de la mejora de la calidad de los diseños, programas y documentos, la programación sin ego también mejora las comunicaciones dentro del grupo. Ésta permite discusiones desinhibidas sin considerar status, experiencia o sexo. Los miembros cooperan con otros miembros del equipo a través del desarrollo del proyecto. Esto permite que los miembros del equipo trabajen juntos y se sientan como parte de un equipo.

La cohesión del grupo depende de muchos factores, entre los que se encuentran la cultura organizacional y las personalidades del grupo. Los administradores pueden fomentar la cohesión de varias formas. Pueden organizar eventos sociales para los miembros del grupo y sus familias. Pueden tratar de establecer un sentido de identidad del grupo asignándole un nombre y estableciendo un territorio. Pueden estar involucrados en actividades en grupo como deportes y juegos.

Sin embargo, una de las formas más eficaces de promover la cohesión es asegurar que los miembros del grupo sean tratados como responsables y confiables y se les dé acceso a toda la información. A menudo, los administradores sienten que no tienen que revelar cierta información a todo el grupo. Esto invariablemente crea un clima de desconfianza. El intercambio de información simple es una forma barata y eficiente de hacer que la gente sienta que es parte del grupo.

Podemos ver un ejemplo de esto en el caso de estudio de la Figura 25.6. Alice realiza reuniones informales regulares donde comenta al grupo lo que está haciendo. Ella involucra a la gente en el desarrollo del producto, preguntándoles sobre las nuevas ideas derivadas de las ex-

Caso de estudio 4. El espíritu del equipo

Alice, una gestora de proyectos experimentada, entiende la importancia de crear un grupo cohesivo. Como el desarrollo del producto es nuevo, aprovecha la oportunidad de involucrar a todos los miembros del grupo en la especificación y el diseño, obteniendo opiniones sobre la posible tecnología a aplicar con los miembros mayores de sus familias, y trayendo a miembros de su familia al almuerzo semanal del grupo. El almuerzo del grupo es una oportunidad para que todos los miembros del equipo se reúnan informalmente, hablen de sus problemas y se conozcan unos a otros.

En el almuerzo, Alice dice a los miembros del grupo lo que sabe sobre noticias organizacionales, políticas, estrategias, etc. Cada miembro del equipo entonces resume lo que había estado haciendo y el grupo habla acerca de temas generales, como las nuevas ideas de producto para sus parientes mayores.

Cada pocos meses, Alice organiza un «día de visitantes» para el grupo, donde el equipo dedica dos días a actualizar tecnología. Cada miembro del equipo prepara una actualización sobre una tecnología relevante y la presenta al grupo. Ésta es una reunión fuera del lugar de trabajo, en un buen hotel, y el tiempo está programado para el diálogo y la interacción social.

Figura 25.6
Cohesión
del grupo.

periencias en su propia familia. Estas reuniones son un buen camino para fomentar la cohesión; las personas se relajan mientras se ayudan mutuamente a aprender las nuevas tecnologías.

Sin embargo, los grupos fuertes y cohesivos algunas veces padecen dos problemas:

1. *Resistencia irracional al cambio de liderazgo.* Si el líder de un grupo cohesivo se reemplaza por alguien externo al grupo, los miembros del grupo se asocian en contra del nuevo líder. Además, dichos miembros invierten tiempo al resistirse a los cambios propuestos por el nuevo líder del grupo, lo que lleva a una pérdida consecuente de la productividad. Por lo tanto, siempre que sea posible, es mejor designar a los nuevos líderes de dentro de los grupos.
2. *Pensamiento de grupo.* El pensamiento de grupo (Janis, 1972) es el nombre que recibe la situación en que las habilidades importantes de los miembros del grupo se corrompen por la lealtad al grupo. La consideración de alternativas se reemplaza por lealtad a las normas y decisiones del grupo. Cualquier propuesta favorecida por la mayoría del grupo se adopta sin consideración de alternativas.

Para evitar el pensamiento de grupo, se pueden organizar sesiones formales donde a los miembros del grupo se les pida que critiquen las decisiones. Se pueden introducir expertos para que revisen las decisiones del grupo. Las personas que argumentan, cuestionan y no respetan el *statu quo* se designarán como miembros del grupo. Actúan como un abogado del diablo, constantemente cuestionan las decisiones del grupo, y así fuerzan a otros miembros del grupo a razonar y evaluar sus actividades.

25.3.3 Las comunicaciones del grupo

Es esencial que exista buena comunicación entre los miembros de un grupo de desarrollo de software. Los miembros del grupo deben intercambiar información sobre el estado de su trabajo, las decisiones de diseño que se han tomado y los cambios necesarios para las nuevas decisiones. Las buenas comunicaciones también fortalecen la cohesión puesto que los miembros del grupo comprenden las motivaciones, fortalezas y debilidades de las personas del grupo.

Algunos factores clave que influyen en la efectividad de las comunicaciones son:

1. *El tamaño del grupo.* Si un grupo incrementa su tamaño, es más difícil asegurar que todos los miembros se comuniquen unos con otros de forma efectiva. El número de vínculos de comunicación en una dirección es $n * (n - 1)$ donde n es el tamaño del grupo. Como se puede ver, con un grupo de siete u ocho miembros, es muy posible que alguna persona raramente se comunique. Las diferencias de status entre los miembros del grupo implican que a menudo las comunicaciones son en una sola dirección. Los miembros de status más alto tienden a dominar las comunicaciones con los miembros de status más bajo, quienes a menudo rehúyen iniciar una conversación o hacer observaciones de crítica.
2. *La estructura del grupo.* Las personas en grupos estructurados informalmente se comunican de forma más efectiva que los grupos con una estructura jerárquica formal. En los grupos jerárquicos, las comunicaciones tienden a fluir hacia arriba y hacia abajo de la jerarquía. Las personas al mismo nivel no hablan unas con las otras. Éste es un problema particular en un proyecto grande con varios grupos de desarrollo. Si las personas que trabajan en diferentes subsistemas sólo se comunican con sus administradores, a menudo esto conduce a retrasos y malentendidos.

3. *La composición del grupo.* Si existen demasiadas personas en el grupo que tienen los mismos tipos de personalidad chocan entre ellas y las comunicaciones se inhiben. Por lo general, la comunicación es mejor en grupos de ambos sexos (Marshall y Heslin, 1975) que en grupos de un solo sexo. Las mujeres tienden a estar más orientadas a la interacción que los hombres y los miembros de un grupo de mujeres facilita y dirige la interacción en el grupo.
4. *El entorno de trabajo físico del grupo.* La organización del lugar de trabajo es un factor importante para facilitar o inhibir las comunicaciones. Esto se analiza en la Sección 25.3.5.

25.3.4 La organización del grupo

Los grupos pequeños de programación, por lo general, se organizan de una forma más informal. El líder del grupo se involucra en el desarrollo de software con los otros miembros del grupo. De esta forma, emerge un líder técnico quien controla de forma efectiva la producción de software. En un grupo informal, el trabajo a realizar se discute por el grupo como un todo y las tareas se asignan de acuerdo con la habilidad y experiencia. Los miembros más experimentados del grupo realizan el diseño de sistemas de alto nivel, pero el diseño de bajo nivel es responsabilidad de los miembros a quienes se les asigna una tarea particular.

Los grupos informales pueden ser muy exitosos, particularmente cuando la mayoría de los miembros del grupo son experimentados y competentes. El grupo funciona como una unidad democrática, que toma decisiones por consenso. Psicológicamente, esto mejora el espíritu del grupo con un incremento en la cohesión y el desempeño. Si un grupo se compone en su mayoría de miembros sin experiencia o incompetentes, informalmente pueden ser un estorbo. No existe una autoridad definida para dirigirse al grupo, lo que provoca una falta de coordinación entre los miembros del grupo y, posiblemente, un desastre para éste.

Beck, en su libro sobre programación extrema (Beck, 2000), describe una variante organizacional interesante de la organización democrática de grupos. En este enfoque, muchas decisiones que por lo general se ven como decisiones administrativas —como las decisiones de calendario— son desarrolladas por los miembros del grupo. Los programadores trabajan en parejas para desarrollar el código y tomar una responsabilidad colectiva para los programas a desarrollar.

Como se explica en el Capítulo 26, la habilidad individual influye de forma importante en la productividad del programador. Los mejores programadores pueden tener una productividad 25 veces mayor que los peores programadores. Por lo tanto, lo mejor será utilizar los mejores programadores y darles todo el apoyo necesario.

Para utilizar de forma más efectiva a los programadores con mucha habilidad, Baker (1972) y otros (Anon, 1974; Brooks, 1975) señalaron que los equipos deben construirse alrededor de un programador en jefe con muchas habilidades. El principio subyacente es que el personal con habilidad y experiencia sea el responsable de todo el desarrollo del software. Ellos no deben estar implicados en cuestiones de rutina, y deben tener un buen apoyo técnico y administrativo en su trabajo. Deben centrarse en el software a desarrollar y no involucrarse en reuniones externas.

Pero la organización del equipo tiene serios problemas porque sobrecarga al jefe de programadores y a su asistente. Otros miembros del equipo, a los cuales no se les ha dado la suficiente responsabilidad, comienzan a desmotivarse debido a que sienten que sus habilidades son infrautilizadas.

A pesar de ello, el principio general de incrementar el equipo de programación con especialistas es una buena idea. Cuando se elija a los miembros del equipo, la elección puede cen-

trarse en gente con habilidades genéricas como comunicaciones y resolución de problemas, y luego se puede incorporar a expertos cuando el proyecto lo requiera. El uso de expertos es necesario, y los desarrolladores inexpertos tienen la oportunidad de aprender y desarrollar su pericia a medida que progresa el proyecto.

25.3.5 Entornos de trabajo

El lugar de trabajo tiene efectos importantes en el desempeño de las personas y en su satisfacción en el trabajo. Los experimentos psicológicos han mostrado que el comportamiento se ve afectado por el tamaño de la habitación, el mobiliario, el equipo, la temperatura, la humedad, la luminosidad y la calidad de la luz, el ruido y el grado de privacidad disponible. El comportamiento del grupo se ve afectado por la organización arquitectónica y los recursos de telecomunicaciones. Las comunicaciones dentro del grupo se ven afectadas por la arquitectura del edificio y la organización del lugar de trabajo.

Existe un costo importante y real para proveer buenas condiciones de trabajo. Cuando las personas no están felices con sus condiciones de trabajo, el movimiento de personal se incrementa. Por lo tanto, tiene mucho más costo el reclutamiento y la capacitación. Los proyectos de software se retrasan debido a la falta de personal cualificado (DeMarco y Lister, 1999).

A menudo, el personal de desarrollo de software trabaja en áreas y oficinas abiertas, algunas veces en cubículos, y sólo los administradores principales tienen oficinas individuales. McCue (1978) llevó a cabo un estudio que muestra que la arquitectura abierta que favorecía a muchas organizaciones no era popular ni productiva. Los factores del entorno más importantes que se identificaron en ese estudio de diseño fueron:

1. *Privacidad.* Los programadores requieren un área donde se puedan concentrar y trabajar sin interrupción.
2. *Repercusión del exterior.* Las personas prefieren trabajar con luz natural y con una vista del entorno exterior.
3. *Personalización.* Los individuos adoptan diferentes prácticas de trabajo y tienen diferentes opiniones sobre la decoración. La habilidad para arreglar el lugar de trabajo que permita llevar a cabo éste y personalizar ese entorno es importante.

En resumen, a las personas les gustan las oficinas individuales que puedan organizar como ellos quieran. Existen menos distracciones e interrupciones que en espacios de trabajo abiertos. En las oficinas abiertas, a las personas se les niega la privacidad y un entorno de trabajo silencioso y se les limita la personalización de su lugar de trabajo. La concentración es difícil y el desempeño se degrada.

Proveer oficinas individuales al personal de ingeniería de software puede tener una diferencia significativa en la productividad. DeMarco y Lister (1985) compararon la productividad de los programadores en varios tipos de lugares de trabajo. Observaron que factores como la privacidad del lugar de trabajo y la habilidad para eliminar las interrupciones tenían un efecto importante. Los programadores que tenían buenas condiciones de trabajo eran dos veces más productivos que los programadores con las mismas habilidades pero que tenían condiciones de trabajo peores.

Los grupos de desarrollo necesitan áreas donde todos los miembros del grupo se puedan reunir como grupo y discutir sus proyectos, tanto formal como informalmente. Las salas de juntas deben tener espacio para acomodar a todo el grupo con privacidad. Los requerimientos de privacidad individuales y los requerimientos de comunicación del grupo son objetivos primordiales. McCue sugirió que agrupar oficinas individuales alrededor de las grandes salas de reunión (véase la Figura 25.7) era la mejor solución.

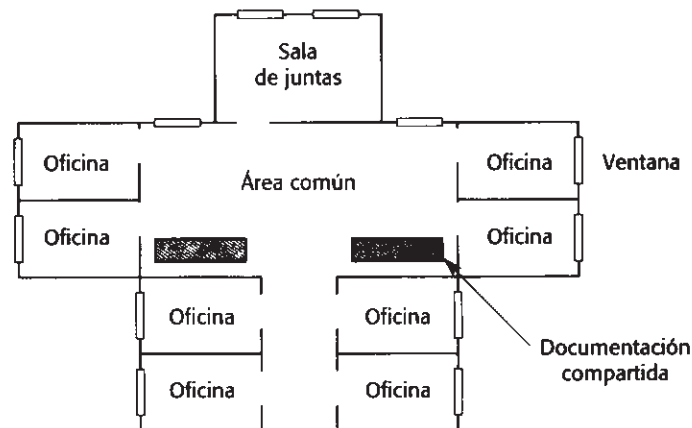


Figura 25.7
Oficina y sala de reuniones del grupo.

Beck (2000) sugirió un modelo similar en su descripción de un entorno para la programación extrema. Sin embargo, sugiere mantener las áreas abiertas, en las que todas las actividades de programación tienen lugar en un área común, y cubículos individuales para los miembros del grupo que deseen trabajar aislados. Claramente, el requerimiento principal es proveer espacios individuales y espacios de grupo para que las personas puedan trabajar solas o en grupo si es necesario.

Este tipo de comunicación ayuda a las personas a resolver sus problemas e intercambiar información de manera informal pero eficaz. Weinberg (Weinberg, 1971) cita un ejemplo anecdótico de cómo una organización quiso hacer que los programadores no desperdiciaran tiempo hablando entre sí alrededor de una máquina de café. Quitaron la máquina, y de forma inmediata tuvieron un notable incremento de solicitudes para asistencia de programación formal. Además de conversar alrededor de la máquina las personas resolvían sus problemas entre sí. Esto ilustra que las compañías necesitan lugares de reunión informales, así como salas de conferencias formales.

El caso de estudio de la Figura 25.8 muestra lo frecuente que es trabajar con restricciones impuestas por los edificios. No es posible adaptarlos o tener todo el espacio que se quisiera. En el ejemplo, Alice usa un despacho individual para trabajos donde se requiere concentración y donde la gente puede discutir en privado qué debe hacer. Los escritorios se comparten

Caso de estudio 5: Organización de la oficina

Alice entiende la importancia de los entornos de trabajo pero su compañía tiene un edificio de los años 70 que no puede ser adaptado a una estructura ideal. Le han asignado tres oficinas para su equipo —una pequeña, individual y separada y dos grandes y juntas, con capacidad cada una para cuatro mesas—. Dos miembros del equipo (Carol y Brian) frecuentemente trabajan desde casa y Fred, el experto en alarmas, sólo trabaja con el equipo dos días por semana. El equipo puede utilizar una sala de reuniones compartida con el resto de los grupos y cada planta del edificio tiene una zona de café para reuniones informales.

En lugar de utilizar la oficina pequeña como su oficina personal, como lo hubiese hecho otro gestor, Alice decide que debe ser una zona tranquila donde el miembro del equipo que lo necesite pueda trabajar sin distracciones. Ella designa una oficina para desarrollo con mesas para el hardware y para los papeles de los prototipos de las interfaces de usuario. Además, esta habitación tiene un escritorio que normalmente usa Fred cuando está trabajando con el equipo, pero también es compartida por Carol y Brian cuando van a trabajar a la oficina. Alice comparte la otra oficina con Bob, Dorothy y Ed. El edificio tiene una red wireless y todos los miembros tienen calculadoras portátiles.

Figura 25.8
Organización de la oficina.

entre los miembros del equipo que no siempre trabajan en la oficina. Cada miembro del equipo tiene una computadora portátil, y puede trabajar en cualquier lugar: en su escritorio, en la habitación tranquila o en zonas comunes del edificio.

25.4 El Modelo de Madurez de la Capacidad del Personal

El Software Engineering Institute (SEI) en Estados Unidos lleva a cabo un programa a largo plazo, de mejora del proceso del software. Parte de este programa es el Modelo de Madurez de la Capacidad (CMM) para el proceso del software que se analiza en el Capítulo 28. Éste se refiere a las mejores prácticas en ingeniería de software. Para soportar este modelo también propusieron un Modelo de Madurez de la Capacidad del Personal (P-CMM) (Curtis *et al.*, 2001). Éste se puede utilizar como un marco de trabajo para mejorar la forma en la que una organización administra sus recursos humanos.

De igual forma que el CMM, el P-CMM es un modelo de cinco niveles como se muestra en la Figura 25.9. Los cinco niveles son:

1. *Inicial Ad hoc*. Se utilizan prácticas informales de administración de personal.
2. *Repetible*. Establece las políticas para el desarrollo de la capacidad del personal.

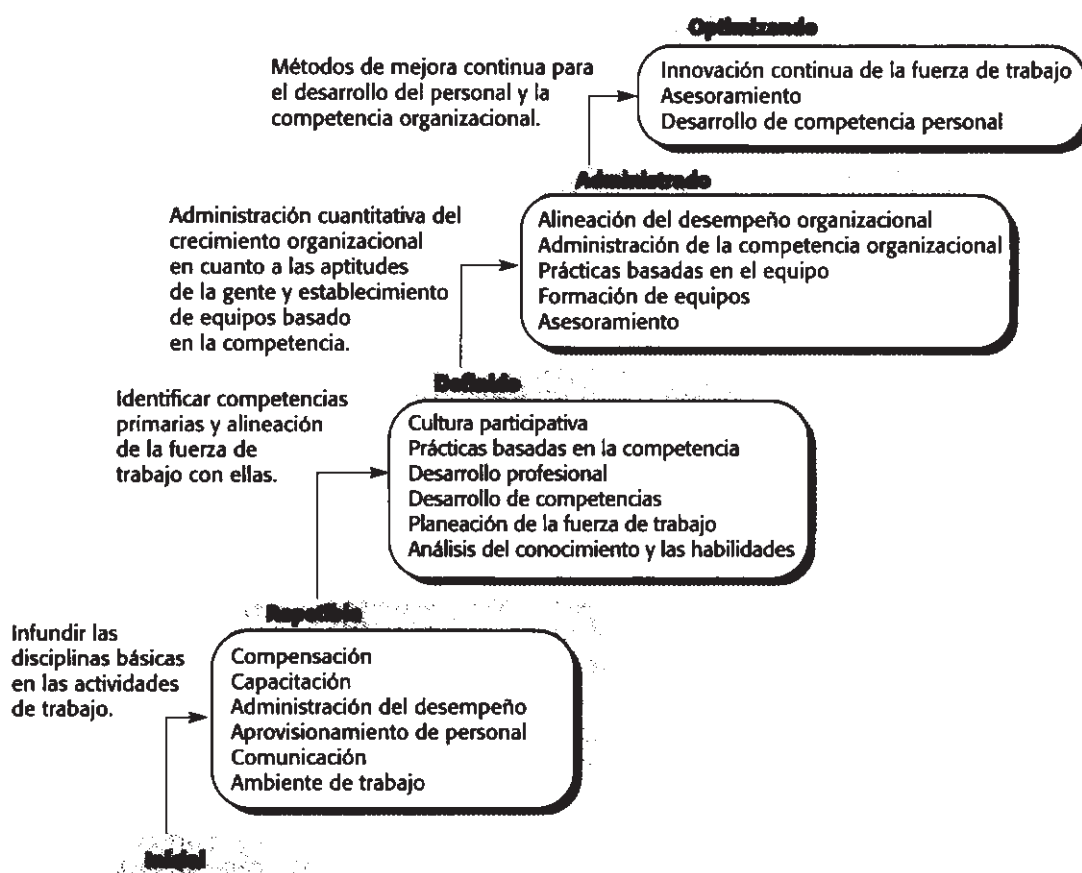


Figura 25.9 El Modelo de Madurez de la Capacidad del Personal.

3. *Definido*. Estandarización de las mejores prácticas de administración de personal a lo largo de la organización.
4. *Administrado*. Se establecen e introducen las metas de la administración del personal.
5. *Optimizado*. Existe un enfoque continuo en la mejora de la competencia individual y la motivación de la fuerza de trabajo.

Curtis *et al.* (Curtis *et al.*, 2001) señalan que los objetivos estratégicos del P-CMM son:

1. Mejorar la capacidad de las organizaciones de software incrementando la capacidad de su fuerza de trabajo.
2. Asegurar que la capacidad de desarrollo de software sea un atributo de las organizaciones más que de unos pocos individuos.
3. Alinear la motivación de los individuos con la de la organización.
4. Retener los activos humanos (por ejemplo, personas con habilidades y conocimiento importantes) dentro de la organización.

El P-CMM es una herramienta práctica para mejorar la administración de las personas en una organización ya que suministra un marco de trabajo para motivar, reconocer, estandarizar y mejorar las buenas prácticas. Refuerza la necesidad de reconocer la importancia de las personas como individuos y de desarrollar sus capacidades. Por supuesto, la aplicación completa de este modelo es muy extensa y probablemente innecesaria para muchas organizaciones. Sin embargo, el modelo es una guía de ayuda que puede conducir a mejoras importantes en la capacidad de las organizaciones para producir software de alta calidad.



PUNTOS CLAVE

- La selección del personal de un proyecto es una de las tareas más importantes de los gestores de proyectos. Algunos de los factores que pueden usarse para la selección de personal son: experiencia en el dominio de la aplicación, adaptabilidad y personalidad.
- La gente es motivada por la interacción con otras personas, por el reconocimiento de los gestores y de sus compañeros, y por las oportunidades de desarrollo personal.
- Los grupos de desarrollo software deben ser pequeños y cohesivos. Los líderes del grupo deben ser competentes técnicamente y deben tener ayuda administrativa y técnica.
- Las comunicaciones dentro del grupo están influenciadas por factores como el status de los miembros del grupo, el tamaño del grupo, la composición sexual del grupo, las personalidades y los canales de comunicación disponibles.
- Los entornos de trabajo para equipos deben incluir espacios donde el equipo pueda interactuar y otros donde los miembros del equipo puedan trabajar individualmente de forma tranquila y concentrándose en su trabajo.
- El Modelo de Madurez de la Capacidad del Personal provee un marco de trabajo y sugerencias para mejorar las capacidades de las personas dentro de una organización y mejorar las capacidades de la organización para obtener los máximos beneficios de sus recursos humanos.

LECTURAS ADICIONALES

A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories. Este libro trata de conclusiones empíricas, hipótesis y teorías relevantes para la ingeniería del software. El Capítulo 10 estudia las habilidades, motivación y satisfacción del usuario y habla de teorías de psicología que refuerza estos temas. (A. Andres y D. Rombach, 2003, Addison-Wesley.)

Software Management, 6th ed. Éste es un tutorial de IEEE que tiene varios artículos sobre cómo dirigir y motivar a las personas. (D. j. Reifer, 2002, Wiley-IEEE Press.)

The People Capability Maturity Model: Guidelines for Improving the Workforce. Este libro hace una descripción fácilmente comprensible del modelo P-CMM, incluyendo una guía para mejorar las capacidades individuales, desarrollar una estructura organizacional fuerte, medir el rendimiento y crear una organización flexible de trabajo. (B. Curtis et al., 2001, Addison-Wesley.)

Peopeware: Productive Projects and Teams, 2nd ed. Éste es un libro clásico sobre la importancia de tratar adecuadamente a las personas cuando gestionamos proyectos de software. Es fácil de leer y es uno de los pocos libros donde se reconoce la importancia del lugar en el que se trabaja. Se recomienda ampliamente. (T. DeMarco y T. Lister, 1999, Dorset House.)

EJERCICIOS

- 25.1** Explique por qué la coherencia, respeto, inclusión y franqueza son factores que contribuyen eficazmente en la gestión de personal.
- 25.2** ¿Qué factores deben tenerse en cuenta a la hora de seleccionar el personal para un proyecto software? Razone la respuesta y sugiera cuáles de éstas serán más importantes en la selección de personal para el desarrollo de un sistema empotrado en tiempo real a fin de desarrollar un controlador para un ojo de una máquina de cirugía.
- 25.3** Desarrolle el caso de estudio del ejemplo sobre motivación de la Figura 25.4 incluyendo actividades generales que Alice podría incorporar para asegurarse de que otros miembros del equipo sigan motivados.
- 25.4** Explique por qué mantener a los miembros del equipo informados acerca del progreso y de las decisiones técnicas del proyecto puede mejorar la cohesión del grupo.
- 25.5** Explique lo que entiende por «pensamiento en grupo». Describa los peligros de este fenómeno y explique cómo pueden evitarse.
- 25.6** Explique qué problemas pueden aparecer en un equipo de programación extrema donde muchas decisiones de gestión son tomadas por los propios miembros del equipo.
- 25.7** ¿Por qué las oficinas de plan abierto y oficinas compartidas son a veces menos recomendables que las individuales para el desarrollo de software? ¿En qué circunstancias piensa que los entornos abiertos pueden ser mejores?
- 25.8** ¿Por qué es P-CMM un marco eficaz para la mejora de gestión de personal en una organización? Sugiera cómo puede ser modificado para ser usado en pequeñas compañías.
- 25.9** ¿Deben los gestores hacerse amigos y mezclarse socialmente con los miembros más jóvenes de su equipo de trabajo?
- 25.10** ¿Es ético suministrar las respuestas que el examinador desea ver más que decir lo que uno siente cuando se llevan a cabo pruebas psicológicas?