

## Capítulo 3

### Asignación de recursos en un proyecto informático

---

3.1. INTRODUCCIÓN .....	2
3.2. DETERMINACIÓN DEL PLAZO DE ENTREGA DE UNA APLICACIÓN.....	3
3.2.1. Identificación de los límites temporales del proyecto versus asignación de recursos .....	4
3.2.2. Determinación del plazo .....	5
3.3. TIPOS DE RECURSOS USUALES .....	9
3.4. DURACIÓN DE LAS TAREAS.....	10
3.4.1. Esfuerzo y duración de las tareas .....	10
3.4.2. Asignación de personas a tareas .....	12
3.4.3. Tipo y duración de tareas en función de la cantidad de personas asignadas .....	13
3.5. ASIGNACIÓN CONSISTENTE DE LAS TAREAS .....	14
3.6. CONSIDERACIONES FINALES .....	15

### **3.1. Introducción**

La asignación de recursos consiste en asociar a cada una de las tareas, en el proyecto, las personas, equipos y materiales necesarios para que éstas se puedan realizar. Ésta es una labor complicada y fundamental en la planificación del desarrollo de una aplicación informática.

La visión de las personas como recursos es errónea. Como dice Handy en “La era de lo irrazonable”: “Las personas no son recursos humanos. Son individuos vivos, con todo su derecho a ser diferentes”. De hecho, Peter Druker, haciendo referencia a las empresas del futuro dice que no se parecerán a lo que pueden imaginar empresarios y profesores, sino que serán similares a los hospitales, universidades o grandes orquestas sinfónicas. Al igual que éstas tendrán en el conocimiento su principal recurso, pasando a ser organizaciones compuestas fundamentalmente por especialistas que trabajarán de acuerdo a las informaciones que reciban.

En cualquier caso esto no eximirá al director o coordinador de tener que enfrentarse al reparto de tareas y adscripción de las mismas a los miembros de la organización

Actualmente los recursos humanos son el componente económico más importante, en los proyectos informáticos, por encima de los recursos físicos como hardware o instalaciones.

El hecho de no tener en cuenta los recursos físicos de instalaciones y hardware necesarios para el desarrollo parece deberse a que el precio del hardware baja de forma continua y que en todo caso el consumo de estos recursos es función de la cantidad de meses-hombre que dure el proyecto.

Otro aspecto importante es el de los consultores, profesionales externos, que asesoran y dan soporte a tareas en donde la empresa no tiene experiencia o le resultaría oneroso mantener a un empleado. En grandes proyectos pueden llegar a suponer un importe similar al consumido por las personas que desarrollan la aplicación.

De forma general podemos afirmar que el coste del proyecto total es el doble del de los recursos humanos, el de los desarrolladores más el del hardware, instalaciones y software

base. En grandes proyectos, que se alejen del proyecto típico de la organización habrá que añadir los servicios de empresas consultoras, que pueden significar una parte más en el coste (tres veces el gasto en recursos humanos). En cualquier caso es importante que la empresa mantenga un registro del esfuerzo dedicado a los proyectos informáticos y su coste global.

En los recursos utilizados por el proyecto habría que incluir el consumo de tiempo dedicado a éste por parte de los usuarios. No se suele tener en cuenta y puede suponer un esfuerzo considerable, aunque no se evalúe. Suele salir a la luz en las críticas del tipo: “¡Con el tiempo que nos han estado consultando y vaya aplicación han hecho!”. También queda patente cuando un usuario justifica el no asistir a una reunión por tener mucho trabajo pendiente.

Desde un punto de vista global, además de las tareas propias del proyecto debemos tener en cuenta, que para que un grupo haga su trabajo, es necesario que:

- Se realicen las tareas en si mismas.
- Se realicen tareas de mantenimiento del equipo, lo que ayuda a mantener su cohesión, su motivación y su voluntad general de dedicarse a la tarea.
- Se satisfagan las necesidades individuales, puesto que ayuda al individuo a sentirse parte del grupo y le capacita para realizar su aportación máxima.

Las planificaciones forzadas artificialmente, para que duren o cuesten menos de lo previsible, condenan al proyecto independientemente de la calidad del personal o de la disponibilidad de herramientas, lenguajes y procesos de desarrollo. Si se trata de comprimir la duración o el presupuesto, el personal que trabaje en el proyecto, no lo hará eficientemente, no se esforzarán si ven que es imposible alcanzar la meta. Peor aún, cuando los retrasos empiecen, sufrirá la moral y el proyecto probablemente cueste más de lo que hubiera costado de haberse hecho de forma razonable.

Como se ve la asignación de recursos es complicada e implica a muchas personas.

## **3.2. Determinación del plazo de entrega de una aplicación**

A lo largo de la asignatura utilizamos los términos “Proyecto Informático” y “Aplicación” como complementarios y a veces como sinónimos. La diferencia entre estos conceptos, para

los informáticos, es que mientras la aplicación es el objetivo, el proyecto es el medio para alcanzarlo. Los clientes y usuarios suelen percibirlo de forma distinta, la aplicación es el producto que les hace falta para poder alcanzar sus objetivos empresariales, y cuanto antes esté disponible mejor. Mientras que el proyecto lo perciben como algo “oscuro”, que siempre consume más de lo que se dijo en dinero y tiempo.

De modo que al determinar el plazo de entrega de la aplicación hay dos variables importantes, que responden a:

- ¿Cuánto tiempo y dinero consumirá este proyecto? y
- ¿Cuándo deberíamos tener disponible la aplicación para optimizar el trabajo de los usuarios?

### **3.2.1. Identificación de los límites temporales del proyecto versus asignación de recursos**

Antes de obtener una planificación definitiva con la asignación de recursos, debemos conocer, a grandes rasgos, las limitaciones temporales del proyecto.

Evidentemente si un proyecto requiere el esfuerzo de 165 meses/hombre y si decidimos que una persona realice el proyecto, dentro de 15 años entregaremos la aplicación al usuario, si la empresa existe y el usuario o su departamento siguen teniendo interés en esta aplicación. Hay varias razones que hacen absurdo el planteamiento anterior:

- 1) Los negocios actuales son muy agresivos lo que implica que las organizaciones que les dan soporte han de ser ágiles y flexibles. Posiblemente ningún sistema pueda sobrevivir 10 años en una empresa.
- 2) Una aplicación importante para la empresa deberá estar disponible lo antes posible dados los costes de oportunidad que tendrá.
- 3) La tecnología evoluciona a tal velocidad que el lenguaje o entorno de trabajo de la aplicación quedarán obsoletos mucho antes de estar disponible la aplicación. Imagine una aplicación que comenzó a desarrollarse hace 15 años con las últimas tecnologías.
- 4) El desarrollo de aplicaciones informáticas, de cierta envergadura, requiere de especialistas en diversas áreas como bases de datos, telecomunicaciones, diseño de interfaces, obtención de especificaciones, programación de sistemas, etc.

Éstas son razones suficientes para pensar que es imposible que una sola persona lleve a cabo el desarrollo de una aplicación como ésta.

Lo opuesto es pensar en desarrollar la aplicación en un sólo día con  $165 * 20 = 3.300$  personas en el desarrollo. Esto no es viable técnicamente, dado que unas tareas no se pueden iniciar antes de la finalización de otras. Alguien puede pensar que asignando muchas personas a una tarea, esta se podrá finalizar en digamos una hora, cosa que como veremos también resulta técnicamente imposible.

Así pues, los proyectos informáticos deberán ajustar su duración:

- Por una parte, adaptándose a los **aspectos del negocio** que nos indican unas fechas a partir de las cuales ya no resulta interesante disponer de la aplicación o tener unos costes de oportunidad elevados.
- Por otra parte, a los **aspectos técnicos** del desarrollo que indican la cantidad máxima de recursos que se pueden asignar a cada tarea.
- Además, los **aspectos de gestión** hacen aconsejable tener un equipo de desarrollo lo más pequeño posible, con el fin de evitar problemas de comunicación y coordinación.

Evaluando estos aspectos, y otros que veremos en la asignatura, se deberá llegar a un consenso sobre la duración total del proyecto y el consumo de recursos a realizar.

### 3.2.2. Determinación del plazo

Para determinar el plazo de entrega se puede seguir tres caminos, uno se basa exclusivamente en la negociación y dos de ellos siguen un método técnico. Empezaremos por el menos recomendable y más difundido.

#### 1. LA NEGOCIACIÓN

La negociación es una de las técnicas más extendidas para fijar los plazos de entrega de aplicaciones informáticas. Esta técnica puede ser muy peligrosa si se producen ciertas circunstancias como:

- Se comienza a negociar sin tener claras las especificaciones del cliente.
- El usuario, con ligeras nociones de las técnicas de desarrollo actuales, fundamentalmente basadas en presentaciones comerciales de distribuidores de herramientas, fuerza a que el plazo negociado sea lo menor posible.

- El usuario tiene la necesidad de disponer de la aplicación lo antes posible.
- El jefe de proyecto tiene que negociar con un usuario de mayor nivel jerárquico.
- El trabajo usual de muchos usuarios es contratar servicios a empresas externas y saben que siempre hay un margen que se puede disminuir.

Con todas estas circunstancias es normal que tras la negociación de los plazos de entrega de la hipotética nueva aplicación se tengan:

- Fuertes niveles de compromiso personal del jefe del proyecto, y
- Escasa participación en la fijación de plazos de los que van a desarrollar la aplicación.

Este marco es el ideal para el fracaso de un proyecto, ya que el desconocimiento de las necesidades del usuario suele hacer que, por lo general, se subestimen. Además el compromiso unilateral del jefe, en estas condiciones, difícilmente será respaldado por sus subordinados.

En cualquier caso, la negociación es habitual en estas circunstancias, por lo que el planteamiento que se propone puede ser más aconsejable. Además, el espíritu comercial es imprescindible en las empresas.

## 2. SELECCIÓN DE UNA ALTERNATIVA

El segundo método se puede ver como una negociación en la que el director del proyecto ha preparado una serie de alternativas y se las ofrece al cliente. Se parte de la obtención de una especificación de lo que desea el cliente y sobre ella se diseñan diferentes alternativas. Aplicando el historial productivo de la empresa, se estima el esfuerzo necesario para cada alternativa.

En una reunión posterior los posibles implicados en el proyecto (analistas, programadores, etc.) llegan a consensos sobre las planificaciones y recursos necesarios. Si la empresa desea un presupuesto, éste se calcula basándose en las planificaciones realizadas. En cualquier caso, se pueden obtener diferentes planificaciones, para un mismo sistema con distintos diseños e incluso un mismo diseño, modificando los recursos asignados. Los presupuestos y planificaciones se presentan al usuario que selecciona el más apropiado al negocio.

Este sistema tiene muchas probabilidades de alcanzar el éxito ya que, primero: se basa en datos históricos de productividad, y segundo: el compromiso de los desarrolladores es alto, ya que formaban parte del grupo que realizó la estimación.

### 3. MÉTODO EMPÍRICO DE PUTNAM Y NORDEN

El tercer método se basa en estudios empíricos que inicialmente realizó Norden (1963), basados en proyectos de IBM. Se basa en el estudio de la duración de los proyectos y la cantidad de personas asignadas cada mes. Éstos le llevaron a identificar la curva de desarrollo, que se muestra en la Figura 3.1. Putnam estudió y refinó esta curva a partir de datos sobre grandes proyectos de la armada americana. En pequeños proyectos esta curva queda distorsionada.

La fórmula a que obedece esta curva es:

$$\text{Esfuerzo\_en\_personas}(t) = 2K a t e^{-at^2}$$

donde:

- Esfuerzo\_en\_personas es la cantidad de meses/persona que se aplican en un momento determinado,
- a es el factor de aceleración, que indica el incremento inicial de la curva,
- K es el esfuerzo total del proyecto, en meses/persona (área de la curva), y
- t es un instante de tiempo a lo largo de la realización del proyecto.

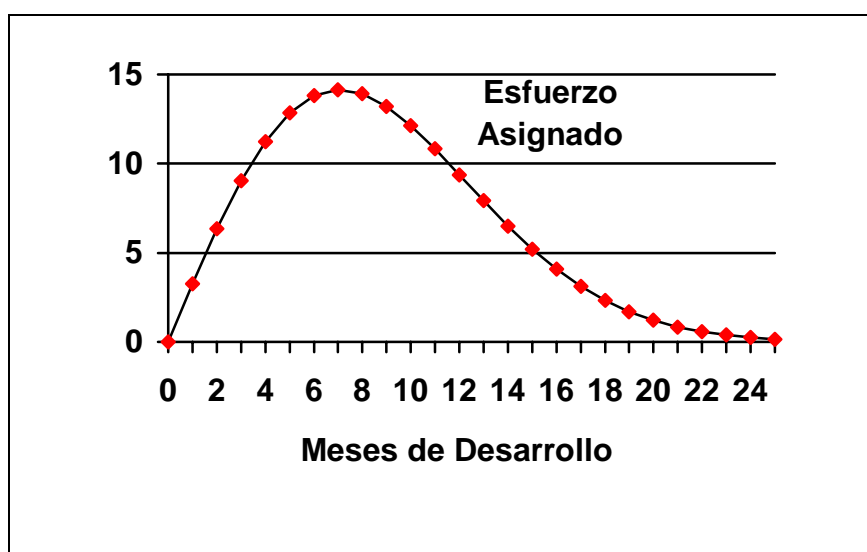


Figura 4.1. Curva para un proyecto de 165 meses/persona

Esta curva contiene la siguiente información:

- Al principio del proyecto aunque se asignen muchas personas, éstas no pueden ser utilizadas de forma eficiente.
- Al ir apareciendo tareas técnicas de programación, diseño e implementación de bases de datos, preparación de pantallas, informes, etc. se puede utilizar más personal.
- Una vez realizado el grueso del trabajo van quedando menos tareas y la cantidad de personas que se pueden asignar son menos.

La curva la podemos aplicar para estimar la duración de un proyecto con K esfuerzo en personas/mes, teniendo unas pérdidas mínimas. Dicho de otra manera, si aplicamos a un proyecto una cantidad fija de personas desde el inicio del proyecto, nos encontraremos en la situación descrita en la Figura 3.2.

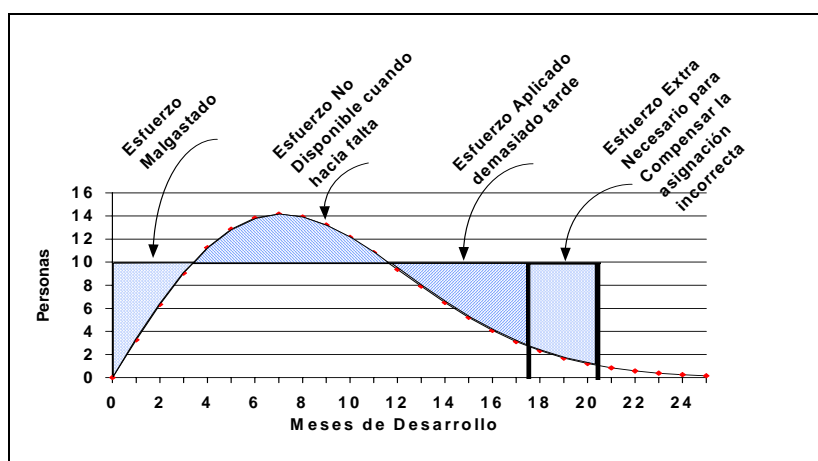


Figura 4.2

Otros usos de estas curvas se dan cuando podemos fijar la cantidad máxima de personas de las que podremos disponer en cualquier momento del desarrollo. En este caso nos podemos mover en una familia de curvas, como las de la Figura 3.3 y seleccionar la que se adapte mejor a nuestra situación.



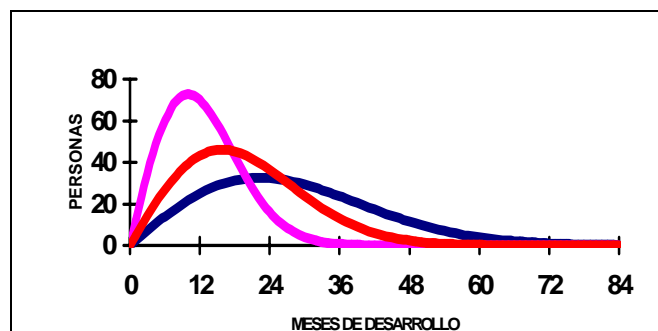


Figura 3.3

Algo similar a esto postuló Boehm, definiendo la región imposible en cuanto a la duración de un proyecto. En concreto, basándose en los recursos necesarios para realizar el proyecto, indica que desde la especificación a la entrega de un producto informático, no puede pasar menos de:

$$T > 2,15 * (\text{personas/mes})^{(1/3)}$$

Y el 99% de los proyectos cumplen esto.

### 3.3. Tipos de recursos usuales

Por recurso entendemos el trabajo de las personas o cosas necesarias para realizar alguna tarea. Podemos clasificarlos en:

- Trabajo, puede realizarse por personas de la organización o externas; podemos descomponerlo en:
  - Equipo de desarrollo: Director del proyecto, Analistas, Diseñadores y Programadores.
  - Soporte al desarrollo: Especialistas en Base de datos, en redes locales y comunicaciones, en interfaces, en formación, así como Operadores y Administrativos.
  - Clientes y Usuarios: Personal de la alta dirección (es recomendable que el liderazgo del proyecto recaiga en una persona de este tipo), Directores y personal de los departamentos afectados.
- Lugar de trabajo, espacio en donde los desarrolladores realizarán sus tareas, podemos clasificarlo en:
  - Salas de reuniones, en donde usuarios, clientes y desarrolladores realizarán tareas conjuntas.
  - Entornos de desarrollo, donde los informáticos realizan trabajos individualmente como documentar o programar. Hay que hacer notar que muchas de las compañías y

empleados más productivos disponen de oficinas silenciosas. Con teléfonos que se pueden silenciar o desviar. Están aislados de las interrupciones que no son propias del asunto.

- Zonas para recogida de datos, lugares de trabajo de los usuarios y zonas de archivos tanto actuales como futuros.
- Equipamiento, muebles necesarios para poder realizar el trabajo, podemos clasificarlo en:
  - Mobiliario de oficina, mesas, sillas, lámparas, teléfono, fax, etc.
  - Material para presentaciones, proyectores, pantallas, mesas apropiadas, etc.
  - Ordenadores, Infraestructura de red, Estaciones de trabajo, Hardware específico del sistema de desarrollo, etc.
- Material básico para el desarrollo, Software y Manuales:
  - Sistemas Operativos, Lenguajes de desarrollo, Herramientas de desarrollo (CASE).
  - Manuales del software: iniciación, manual de usuario, librerías, etc.
  - Libros con referencia a técnicas de desarrollo, por ejemplo, libros sobre UML.
- Material fungible, es el material que se consume durante el desarrollo, por ejemplo:
  - El material de escritorio: bolígrafos, clips, grapas, barras de pegamento, líquido corrector, etc.
  - El material necesario para los equipos: tinta o toner de impresora, papel de impresora, transparencias, *pendrives*, etc.

Describir con tanto detalle algunos de los materiales se debe a que en la práctica se pierde mucho tiempo, de personal de desarrollo, por no estar disponibles cosas que aparentemente tienen poca importancia.

### **3.4. Duración de las tareas**

Por lo visto hasta ahora, las tareas ya tienen asignado un esfuerzo, ahora hay que decidir cuántos recursos se le asignan a cada tarea, de modo que obtengamos la duración en tiempo de cada una de ellas.

#### **3.4.1. Esfuerzo y duración de las tareas**

Normalmente se suele confundir el esfuerzo necesario para desarrollar una tarea (meses/persona) con el tiempo necesario para llevar a cabo la misma.

Una tarea que tenga asignado un esfuerzo de 10 días, puede llevarse a cabo en el plazo de 5 semanas, asignando una persona que trabaje 2 días a la semana. También nos podemos encontrar con el caso contrario en que una tarea que necesita el esfuerzo de 10 días se ejecute en el plazo de 1 semana, empleando a dos personas que trabajen 5 días a la semana.

El esfuerzo aplicado puede verse afectado por interferencias externas. Por ejemplo, teniendo que realizarse un esfuerzo de una semana - persona, y asignando una persona a esta tarea durante una semana, puede no finalizar a causa de las interferencias. Thomsett dice que hay muchas razones para esto y en concreto menciona algunas como:

- Es necesario repetir trabajos o corregir defectos de tareas previas, que se suponían terminadas, para poder realizar la tarea actual.
- Hay que tener en cuenta las vacaciones, fiestas, fiestas locales, puentes, etc. que puedan afectar al proyecto tanto en los lugares de trabajo como en las zonas de clientes o usuarios.
- Otros equipos de la empresa pueden realizar consultas al personal del proyecto actual sobre temas ajenos a éste.
- La falta de administrativos puede implicar que los programadores tengan que realizar todos sus papeleos que deberían haber sido delegados.
- Falta de formación y adiestramiento adecuada en el personal del proyecto.
- Falta de reuniones del equipo.
- Interrupciones de todo tipo como llamadas telefónicas etc.
- Tiempo de espera en reuniones.
- Tiempo que tarda el personal en cambiar de tarea, no se puede esperar que sea instantáneo.

Estos factores se pueden llevar entre el 30% y el 50% del esfuerzo realizado. No sería extraño que para una reunión en otra ciudad, de dos días de duración, una persona dedique cuatro días de esfuerzo si no tiene soporte administrativo y tiene que gestionarse los billetes, las reservas de hotel, etc. Otro ejemplo lo dan los retrasos a la hora de comenzar una reunión por falta de personas.

Aunque parezca paradójico, por culpa de estos factores, las personas con mayor nivel de experiencia y responsabilidad en la empresa son los más afectados. Pues:

- deben enseñar y adiestrar al personal del proyecto en temas no previstos,
- son consultados por otros proyectos, y
- se les suele pedir que asistan a reuniones, presentaciones, etc. que en principio no tienen relación con el proyecto actual.

### **3.4.2. Asignación de personas a tareas**

La experiencia indica que es mejor disponer de un equipo pequeño de buenos profesionales que uno mayor de personas no tan capacitadas. De hecho la gente correcta, aún con herramientas, lenguajes y procesos insuficientes, pueden tener éxito. Lo contrario parece imposible. Por otra parte si confiamos todo a unas pocas personas nos podemos encontrar con problemas: ¿Qué ocurre si se van? De modo que tendremos equilibrar al personal.

A la hora de asignar personas a cada tarea nos encontramos con un grupo de tareas y otro de personas. Al comienzo del tema ya dijimos que las personas no pueden verse como un recurso más. Los enfoques actuales relativos a esta asignación pasan por evaluar de cada empleado y tarea los siguientes aspectos:

- El cognitivo o KAS (*Knowledge, Abilities, Skills*), se puede ver como la capacidad técnica. Es decir:
  - Los conocimientos para realizar la tarea,
  - La capacidad de realizarla, y
  - La experiencia sobre la materia.
- El conativo o MAC (*Motivation, Attachment, Confidence*), se puede ver como la voluntad de realizar la tarea. Es decir:
  - La motivación de la persona,
  - El compromiso que asumirá, y
  - La seguridad que tiene en si para realizarla.

Da la impresión, de que basándose en esto, Fergus O'Connell compara un proyecto con un puzzle e indica que se ha de asignar las piezas (tareas) a las personas. Dice que eres una persona con suerte si encuentras la persona apropiada a cada pieza. Por persona hace referencia al conjunto de personalidad, habilidades, experiencia, motivación, metas personales, puntos fuertes y puntos débiles que nos hacen como somos. Además asocia los

aspectos cognitivos a que la persona pueda realizar la tarea desde el punto de vista técnico. Los aspectos conativos los asocia a si la persona quiere realizar la tarea.

Dado que tenemos una lista de tareas y una lista de personas, se trata de realizar la mejor asignación posible. O'Connell propone las siguientes posibilidades para cada par tarea-persona:

- Puede realizar el trabajo y quiere realizarlo.
  - Esto es lo ideal.
- Puede realizar el trabajo y accede a realizarlo.
  - Está bien para el proyecto, habría que motivar a la persona y buscar, a largo plazo, trabajos que se le ajusten.
- Puede realizar el trabajo, pero no esta dispuesto a realizarlo.
  - Tenemos problemas, hay que ver cuál es el problema y resolverlo o nos encontraremos realmente en la última categoría.
- Puede ser formado para realizar el trabajo.
  - Si la empresa está dispuesta a:
    - gastar dinero y tiempo en formación,
    - modificar la programación del proyecto con la formación,
    - soportar una sobrecarga en la dirección, y
    - a afrontar el riesgo de que no funcione bien,
  - Entonces todo está bien, pues en el futuro es posible que la empresa tenga una persona en sus proyectos de la primera categoría mencionada.
- No puede realizar el trabajo.
  - Tenemos un problema serio, y habrá que identificar otras tareas para esta persona.

### **3.4.3. Tipo y duración de tareas en función de la cantidad de personas asignadas**

En un principio, asignar muchas personas a una tarea no quiere decir que la tarea forzosamente dure menos tiempo. Es decir, la gente y la duración no son intercambiables. Frederick P. Brooks en su libro “*The mythical man-month*” ofrece cuatro tipos posibles de tareas. En cada una de ellas la relación existente entre meses y personas es diferente. Las posibles situaciones son:

- 1) Las tareas se pueden repartir de forma perfecta, sin necesidad de comunicación entre las personas. Cuantas más personas se asignen a una tarea, ésta tendrá una duración proporcionalmente menor.
- 2) La tarea no se puede partir. (Para que nazca un niño se requieren nueve meses, no importa cuantas mujeres se asignen). Este tipo de tareas suelen darse cuando tenemos limitaciones en algún recurso. Así, por ejemplo, si para realizar las pruebas me hace falta un dispositivo hardware especial, del que tan solo se dispone de una unidad, no tendrá sentido que asigne muchas personas a dicha tarea.
- 3) La tarea se puede partir, pero se requiere comunicación entre las personas. En principio es la situación más habitual, dado que de contener subtareas independientes, esto se hubiera tenido en cuenta en la descomposición en tareas y tendríamos varias tareas bien definidas en el proyecto. Por otra parte, es habitual tener tareas grandes que son críticas que requirieren mucho esfuerzo y que necesitamos que tengan una duración limitada.
- 4) La tarea se puede partir pero las interrelaciones son tan complejas que cuesta más tiempo realizar la tarea con muchas personas. Son las tareas en que habitualmente alguien dice: “mira prefiero hacerlo solo, puesto que entre varios no terminaremos nunca”.

### **3.5. Asignación consistente de las tareas**

Uno de los problemas clásicos que se producen al asignar las tareas es que éstas no son vistas de la misma forma por el director y los subordinados. Es muy habitual que si preguntamos a un director qué es lo que espera de un subordinado y lo escribimos en un papel y hacemos lo propio con el subordinado, obtengamos dos pedazos de papel, cuyo único parecido sea el hecho de que son de papel.

Otro problema clásico es que aun cuando hacemos la asignación de tareas de forma cautelosa, muchos trabajadores sienten que alguna de las tareas que se han asignado a otro, le correspondía a él, por lo que la suele reclamar y puede llegar a suponer un conflicto que se encone, según se vaya desarrollando el proyecto.

Una forma de evitar ambos problemas consiste en dar a conocer el plan, de forma más o menos completa a todos los trabajadores, luego se elabora una lista de objetivos y tareas junto a cada trabajador. De este modo se evita el primer problema, además de sernos útil para verificar que el trabajador está conforme con la asignación. Esta lista no debería ser superior a

una pagina, legible en menos de un minuto. Una vez terminada la asignación de objetivos y tareas, recibiremos una serie de solicitudes, quejas y aclaraciones, sobre: las tareas que una persona cree que debería tener asignadas, dado que están muy relacionadas con sus objetivos y que han sido asignadas a otro; las tareas que alguien piensa que no han sido asignadas; e incluso las asignaciones que comparten varios participantes del proyecto, pero que parecen incompatibles.

Pasados un par de días de las quejas, se hace una reunión con todos los participantes, se habla sobre los problemas surgidos. Se reescriben las asignaciones y se entregan de nuevo a los participantes. Este proceso puede requerir un par de iteraciones y las reuniones puede que sean tensas. Pero habrá valido la pena, ya que tendremos una descripción clara y compartida de los objetivos, responsabilidades y tareas asignadas a cada miembro del equipo. Además las tareas no se superpondrán y completarán todas las necesidades del proyecto, tal como se indica en “*Managing a Programming Project*” de Metzger.

### 3.6. Consideraciones finales

Como se ve habrá que buscar la situación en que se optimicen cualquiera o todas estas condiciones:

- Coste mínimo de desarrollo
  - En tiempo
    - Emplear especialistas ya formados en cada área de trabajo, tantos como se pueda, siempre y cuando no se tengan que relacionar de forma compleja por razón de las tareas asignadas.
  - En dinero
    - Utilizar el personal necesario para que se lleven a cabo las tareas, de modo que se deban de comunicar lo mínimo posible y que ya conozcan las áreas que se les asignan.
- Coste mínimo a largo plazo
  - Pensar en el mantenimiento y en otros proyectos
    - Hacer que el personal menos experimentado trabaje en el desarrollo, dando formación en caso necesario.

- Hacer que el personal se sienta promocionado, detectando los objetivos de cada empleado y haciendo que cada nuevo proyecto sea un paso en la consecución de sus objetivos.

Conviene recordar, al asignar personas a tareas, que la productividad de los programadores es muy variable. En un estudio se dio el mismo programa a desarrollar a varios programadores obteniéndose diferencias de 1 a 26 en los niveles de productividad. Esto quiere decir que en las tareas críticas convendrá poner al personal con mayor experiencia y reputación, ya que se espera que sean los más productivos.

## REFERENCIAS

1. Brooks, F.P., "The mythical man-month".
2. DeMarco, T., *Controlling Software Projects*, Prentice Hall.
1. O'Connell, F., *How to run successful projects*, Prentice Hall.
3. Metzger, P. y Boddie, J., *Managing a programming project: people and processes*, Prentice Hall.
4. Thomsett, R., *Third Wave Project Management*, Prentice Hall.
5. Yourdon, E., *Análisis Estructurado Moderno*, Prentice Hall.