

Gestión y control de proyectos informáticos. Estimación de recursos. Planificación temporal y organizativa. Seguimiento.

TEMA 59

ABACUS NT

Índice

- 1. *Introducción***
- 2. *Gestión del proyecto de desarrollo de software.***
- 3. *Definición general del proyecto.***
 - 3.1. Determinación del marco inicial del proyecto.**
 - 3.2. Estrategia de desarrollo de software.**
 - 3.3. Recursos y estimación.**
 - 3.3.1. Recursos humanos.**
 - 3.3.2. Recursos de hardware.**
 - 3.3.3. Recursos de software.**
 - 3.4. Estimación del coste y el esfuerzo.**
- 4. *Planificación***
 - 4.1. Análisis de riesgos.**
 - 4.2. Planificación temporal y asignación de recursos.**
 - 4.2.1. Relación recursos / plazo.**
 - 4.2.2. Definición de fases y tareas. Secuencia, dependencias y paralelismos.**
 - 4.2.3. Distribución de esfuerzos.**
 - 4.2.4. Métodos cuantitativos de planificación temporal.**
- 5. *Seguimiento.***
- 6. *Cierre del proyecto.***
- 7. *Conclusión.***
 - 7.1. Relación con el sistema educativo**
- 8. *Bibliografía***

1. Introducción

Como cualquier empresa humana, los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo.

Esto también se conoce como el Triángulo de la Gestión de Proyectos, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. La restricción de tiempo se refiere a la cantidad de tiempo disponible para completar un proyecto. La restricción de coste se refiere a la cantidad presupuestada para el proyecto.

La restricción de alcance se refiere a lo que se debe hacer para producir el resultado final del proyecto. Para conseguir la ejecución del proyecto cumpliendo dichas restricciones, se realiza un proceso denominado gestión y control de proyectos.

2. Gestión del proyecto de desarrollo de software.

Independientemente del ciclo de desarrollo de software que se utilice, desde un punto de vista abstracto y conceptual las fases genéricas por las que ha de transcurrir un proyecto de desarrollo del software son:

- Definición
- Desarrollo
- Mantenimiento.

En la fase de definición se responde a la pregunta ¿qué desarrollar? Se trata pues de determinar los requerimientos clave del sistema. Esta fase se puede dividir en tres subfases:

- Análisis del sistema
- Análisis de requerimientos.
- Planificación del proyecto

De todas ellas, en este tema nos vamos a centrar en la **planificación del proyecto**.

Planificar un proyecto consiste en la asignación y estimación de recursos, en la definición de las tareas que se han de realizar y en la planificación del trabajo que conlleven.

Estas fases de carácter genérico, se complementan con varias actividades encaminadas a asegurar el éxito del proceso. En este sentido será preciso efectuar abundantes revisiones, con el fin de asegurar que se obtiene la calidad prevista.

También se ha de comprobar que el proceso se documente adecuadamente, para que toda la información sobre el sistema y el software esté disponible en el momento que se requiera.

Otra actividad importante es el control de cambios, o como se denomina ahora, control de versiones. En efecto, es importante que los cambios que se lleven a cabo queden perfectamente registrados, de manera que, si se observa un empeoramiento tras el cambio, todo el proceso pueda ser retrocedido a un nivel inferior.

A la fase de planificación se la suele conocer como **gestión del proyecto de desarrollo de software**, denominación que sustituye ventajosamente a la palabra **planificación**.

La función de gestión ha de ser complementada con un conjunto de actividades de control, seguimiento y evaluación.

Luego las fases son:

Estudio de viabilidad:

Apreciaciones sobre el alcance exacto del proyecto y sobre las necesidades futuras de recursos. Esta fase es importante y depende en gran medida de la experiencia anterior de los encargados de realizarla. Se considera que la estimación es más un arte que una ciencia, pero, de cualquier modo, no puede llevarse a cabo de forma descuidada.

Planificación:

La mayoría de las organizaciones de desarrollo de software se encuentran en nivel mínimo de madurez de su capacidad productiva, lo que quiere decir que el caos y la improvisación dominan sobre la medida y la planificación.

Control y seguimiento:

Es posiblemente la fase más extensa, pues comienza cuando el proyecto se inicia y no culmina hasta su finalización.

Consiste en la evaluación continua del desarrollo del proyecto con el fin de detectar las desviaciones sobre lo planificado y actuar en consecuencia.

Evaluación y cierre del proyecto.

El objetivo de esta actividad es la conclusión del proyecto, registrando la información que se haya establecido, generando un balance final del proyecto y, procediendo a su finalización y cierre.

Esta fase final permite valorar los resultados obtenidos a partir de las distintas actividades del proyecto.

3. Definición general del proyecto.

3.1. Determinación del marco inicial del proyecto.

El objetivo de esta tarea es la definición inicial que permita abrir y aprobar el comienzo de un proyecto, estableciendo la definición general del mismo.

La determinación del marco inicial del proyecto debería contener al menos los siguientes apartados:

Breve descripción general del proyecto:

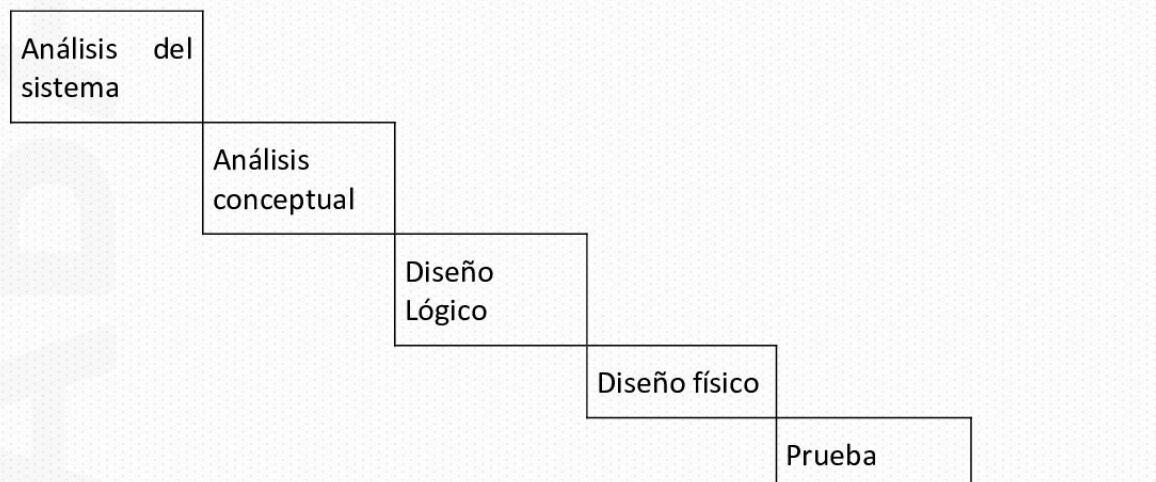
- Objetivos del proyecto: una descripción precisa del proyecto, tanto desde el punto de vista técnico como organizativo.
- Ámbito y alcance, tanto de las áreas organizativas afectadas, como de la naturaleza de los cambios propuestos.
- Establecimiento de los hitos y procedimientos de control, seguimiento y revisión del proyecto.
- Legislación relevante
- Participantes del proyecto y otros grupos y organizaciones
- Planificación y organización inicial del equipo de trabajo.
- Preanálisis de riesgos potenciales.
- Primera estimación de planificación y costes
- Proyectos interrelacionados y concurrentes

3.2. Estrategia de desarrollo de software.

La estrategia de desarrollo, también conocida como **paradigma del ciclo de vida** del proyecto del software, establece cómo debe organizarse el proyecto, mientras que la metodología especifica las tareas, repartidas en actividades y procesos, así como los productos, del ciclo completo de desarrollo del software.

Es posible escoger entre diferentes estrategias:

Secuencial o en cascada: se considera el proyecto como un todo, dividido en procesos, y cada proceso no comienza hasta que finaliza el anterior.



Prototipado y modelo en espiral: esta aproximación genera un prototipo funcional en las primeras fases del proyecto, generalmente con herramientas de ayuda al desarrollo.

El prototipo se completa en sucesivas evaluaciones y revisiones, añadiendo nuevas funcionalidades y mejoras hasta cubrir los requisitos completamente

Prototipado espiral con RAD, si combinamos adecuadamente el prototipado con las herramientas de 4 generación obtenemos lo que se ha dado en llamar técnicas RAD. La utilidad de las mismas consiste en que incrementan la potencia de las herramientas de diseño de prototipos, proporcionando a los desarrolladores todo lo que necesitan para construir un prototipo y también para convertirlo en una aplicación completamente funcional.

3.3. Recursos y estimación.

3.3.1. Recursos humanos.

El planificador comienza evaluando las habilidades técnicas que se requieren para llevar a cabo el desarrollo.

Hay que especificar tanto el puesto dentro de la organización (responsable de evaluación, ingeniero de software..) como la especialidad (bases de datos, redes, programación..).

El número de personas requeridas para un proyecto sólo puede ser estimado con ciertas garantías de acierto después de hacer una previsión del esfuerzo del desarrollo (horas programador).

3.3.2. Recursos de hardware.

Durante la planificación del proyecto de software se deben considerar principalmente dos categorías de hardware: el sistema de desarrollo y el entorno de explotación.

El sistema de desarrollo está compuesto por la computadora que se utilizará durante la fase de desarrollo del software, junto con sus equipos periféricos asociados.

El entorno de explotación es aquel en el que se ejecutará la aplicación una vez que esté completamente terminada y correctamente instalada. Es evidente que habrá que diferenciar adecuadamente las necesidades de ambos tipos, puesto que durante la fase de desarrollo solamente será necesario contar con el primer tipo de hardware, mientras que para que dé comienzo la instalación tendrá que estar disponible la plataforma de ejecución.

3.3.3. Recursos de software.

Desde siempre y cada vez con más intensidad, se viene utilizando software en el desarrollo de los nuevos programas. En este sentido, serán necesarias tanto las herramientas de desarrollo, tal que

entornos de programación, herramientas CASE...etc., como los elementos de software ya desarrollado que se reutilicen en el proyecto.

3.4. Estimación del coste y el esfuerzo.

La estimación de recursos, costes y plazos para el esfuerzo de desarrollo de software requiere indudablemente de experiencia, pero también es fundamental disponer de una buena base de datos con información sobre proyectos anteriores.

Un elemento inherente a la estimación es el riesgo de fallo.

Con independencia de cuan elevada sea la aptitud del equipo encargado de la gestión del proyecto, este riesgo varía normalmente con una serie de factores:

La complejidad del proyecto: tiene un gran efecto sobre la incertidumbre, depende mucho del equipo especializado.

El tamaño del proyecto: se trata de otro factor importante que puede afectar a la precisión y a la eficacia de las estimaciones.

A medida que aumenta el tamaño la interdependencia entre los distintos elementos de software crece rápidamente.

La descomposición del problema, se hará más difícil, dado que los elementos descompuestos pueden seguir siendo todavía de gran tamaño.

La disponibilidad de una B.D. con información sobre desarrollos anteriores: este es un factor fundamental de cara a la reducción del riesgo de la estimación.

Entre estos métodos destacan las técnicas de descomposición, que aplican la técnica de “divide y vencerás” y los modelos empíricos de estimación que utilizan la experiencia acumulada en multitud de proyectos para ayudar al equipo de planificación a prever adecuadamente las necesidades del proyecto en curso.

4. Planificación

La estimación proporciona al grupo planificador del proyecto la información necesaria para llevar a cabo las restantes actividades de la planificación.

4.1. Análisis de riesgos.

El objetivo de esta tarea es identificar y analizar los riesgos potenciales y reales que pueden afectar al proyecto y sus objetivos, así como proponer las medidas oportunas para minimizar o eliminar los posibles inconvenientes que puedan ocasionar dichos riesgos.

El análisis de riesgos se compone de varias actividades diferentes:

- ✓ Identificación y valoración de riesgos, incluye la identificación, descripción y valoración sistemática de los mismos.
- ✓ Planificación de la gestión del riesgo, se identifican y proponen opciones alternativas para la gestión del riesgo.
- ✓ Supervisión de riesgos, se realiza un seguimiento de los factores de riesgo y de las medidas correctivas adoptadas.

4.2. Planificación temporal y asignación de recursos.

El objetivo de esta tarea es la programación detallada del proyecto, planificando en el tiempo la estructura de actividades y tareas, y realizando la asignación de recursos necesaria en función de los distintos perfiles implicados.

Asimismo, se establecen los hitos o puntos de control precisos para la gestión y seguimiento de la planificación detallada.

La planificación puede aplicarse a dos escenarios diferentes. En la primera, la fecha final de lanzamiento del sistema ya ha sido irrevocablemente establecida. El equipo de desarrollo del software se ve forzado a distribuir el esfuerzo dentro del marco prescrito. El segundo enfoque asume que se han estudiado unos límites cronológicos aproximados pero que la fecha es fijada por el propio equipo de desarrollo.

4.2.1. Relación recursos / plazo.

Existen fundamentalmente dos tipos de recursos: humanos y materiales, y un factor limitador : el plazo de realización.

Por recursos humanos entendemos toda aportación de trabajo personal, necesaria para la realización de la aplicación.

Es importante conocer la relación existente entre el plazo de realización y el número de personas implicadas en la aplicación, relación que presentada en forma de curva y en ausencia de otros factores adopta usualmente la forma de una hipérbola, con dos asíntotas una de las cuales tiende hacia un denominado “plazo mínimo”.

En cuanto a recursos materiales también inciden en el plazo final, de manera que a mayor cantidad de recursos menor plazo. No obstante, también existe un límite práctico, por encima del cual será inútil seguir incrementando los recursos materiales, ya que simplemente el equipo humano será incapaz de aprovecharlos, quedando recursos ociosos.

4.2.2. Definición de fases y tareas. Secuencia, dependencias y paralelismos.

Un proyecto completo de desarrollo de software contará con un conjunto de fases cada una de las cuales se podrá dividir a su vez en tareas.

Las fases son las ya conocida (análisis, diseño...) y su estructura y secuencia, e incluso su división en subfases dependen de la estrategia de desarrollo de software que se siga.

Una vez definidas las tareas hay que estudiar las relaciones que se dan entre ellas.

En general existen tareas totalmente independientes, que se pueden realizar en cualquier momento sin considerar al resto de las tareas y, sin embargo, otras que nos pueden iniciarse hasta que las otras de las cuales depende no alcancen cierto nivel de ejecución.

4.2.3. Distribución de esfuerzos.

En cualquier organización típica lo normal es trabajar en un ambiente de recursos limitados. Por este motivo, para planificar un proyecto resulta tan importante conocer las tareas que se han de llevar a cabo, como los recursos que se pueden asignar a las mismas, proceso al que se conoce como **distribución de esfuerzos**.

En el caso de desarrollo de software estos recursos serán principalmente humanos.

4.2.4. Métodos cuantitativos de planificación temporal.

Realizar la planificación temporal es una actividad compleja que requiere del manejo de innumerables datos sobre tareas, sus relaciones, recursos asignados, estimación de plazos, costes, etc.

El objetivo de estos métodos es obtener la programación detallada de actividades del proyecto, definiendo el momento en el que van a realizarse dichas tareas, y asignándole recursos para su ejecución. Se establecen además unos hitos o puntos de control para realizar el seguimiento de la misma.

Para ello es necesario realizar una división del proyecto en una serie de fases y tareas, definir los recursos necesarios para su realización (utilizando técnicas de estimación) y obtener las dependencias entre las taras.

El análisis y diseño de sistemas involucra muchos tipos de actividades que en conjunto forman un proyecto. El analista de sistemas debe administrar el proyecto con cuidado si quiere que tenga éxito. La administración de proyectos incluye las tareas generales de planeación y control.

La planeación incluye todas las actividades requeridas para seleccionar un equipo de análisis de sistema, asignar miembros del equipo a los proyectos apropiados, estimar el tiempo requerido para completar cada tarea y programar el proyecto de manera que las tareas se completen a tiempo. El control implica utilizar retroalimentación para supervisar el proyecto, incluyendo una comparación del plan para el proyecto con su evolución actual. Además, el control implica tomar la acción apropiada para agilizar o reprogramar las actividades de manera que se puedan terminar a tiempo, a la vez que se motiva a los miembros del equipo para que completen el trabajo en forma apropiada.

Diagrama de Gantt

Los diagramas de Gantt son un tipo de gráfico que permite la representación de la localización temporal de las actividades e hitos de un proyecto de desarrollo de software. Tienen forma tabular, correspondiendo las columnas a unidades temporales (día, semana, mes..) y cada fila una actividad.

El nombre de la actividad se sitúa a la izquierda de la tabla y, a su derecha en la misma fila, se dibuja un segmento de barra que va desde la fecha de comienzo a la de finalización de la actividad.

Los diagramas de Gantt también se pueden emplear para el seguimiento de proyectos estimados previamente. Para ello utilizan dos tipos de barras, u otro tipo de señal, de manera que se pueda distinguir las fechas de comienzo y finalización previstas, las fechas de comienzo y finalización reales y, en su caso, el estado de avance de las tareas no concluidas.

La principal ventaja del gráfico de Gantt es su simpleza. El analista de sistemas no sólo descubrirá que esta técnica es fácil de usar, sino que también se presta para una comunicación valiosa con los usuarios finales. Otra ventaja de usar un gráfico de Gantt es que las barras que representan actividades o tareas se dibujan a escala; es decir, el tamaño de la barra indica la longitud de tiempo relativa que se requerirá para completar cada tarea.

Diagramas PERT

PERT es un acrónimo que significa Técnicas de evaluación y revisión de programas. Un programa (sinónimo de proyecto) se representa mediante una red de nodos y flechas que después se evalúan para determinar las actividades críticas, mejorar el programa de trabajo en caso de ser necesario y revisar el avance una vez que se empiece el proyecto. PERT se desarrolló a finales de la década de 1950 para usarlo en el proyecto del submarino nuclear Polaris de la Marina de los EE.UU. Según se informa, ahorró a la Marina de los EE.UU. dos años de tiempo de desarrollo.

PERT es útil cuando las actividades se pueden realizar en paralelo en vez de hacerlo en secuencia. El analista de sistemas se puede beneficiar de PERT al aplicarlo en los proyectos de sistemas a una escala menor, en especial cuando algunos miembros del equipo pueden trabajar en ciertas actividades al mismo tiempo que otros compañeros trabajan en otras.

Existen muchas razones para utilizar un diagrama PERT en vez de un gráfico de Gantt. El diagrama PERT permite:

1. Identificar con facilidad el orden de precedencia.
2. Identificar con facilidad la ruta crítica y, en consecuencia, las actividades críticas.
3. Determinar con facilidad el tiempo de inactividad.

Inicialmente para formar una red PERT sobre un proyecto de desarrollo de software se han de organizar las tareas componiendo un grafo acíclico dirigido.

Para ello, primeramente, hay que identificar las tareas y tiempos asociados a cada actividad. Cada actividad es una parte independiente y homogénea del proyecto que una vez comenzada se realiza de forma independiente del resto de las del proyecto que le preceden y le siguen.

Cada actividad conduce a la obtención de un resultado tangible que puede ser relevante para el desarrollo de otras actividades del proyecto.

Además del concepto de actividad, otros elementos importantes del modelado PERT son los de sucesos externos y los hitos:

- Un suceso externo corresponde a la ocurrencia de una información externa que señala la disponibilidad de un recurso generado en el exterior del proyecto. Tiene el mismo sentido que la finalización de una actividad de la red, esto es, la realización de un suceso condición el desencadenamiento de las actividades que le siguen.
- Un hito es un momento importante del proyecto, como, por ejemplo, la finalización de una etapa relevante, la producción de un resultado significativo para el proyecto o la entrega de algún componente.

5. Seguimiento.

El objetivo de esta actividad es la ejecución y seguimiento del proyecto, supervisando, vigilando y registrando en detalle el cumplimiento de lo establecido en la planificación.

Para ello se llevarán a cabo informes del progreso parcial, y se establecerán los mecanismos oportunos para, conforme se vayan alcanzando los hitos especificados en la definición del proyecto, se apliquen las medidas correctoras que sean necesarias actualizándose, en su caso, las acciones planificadas.

6. Cierre del proyecto.

Cerrar un proyecto es concluirlo, lo que se realizará siempre registrando toda la información relevante que pueda ser utilizada en posteriores planificaciones de proyectos.

Esta actividad de registro de la información real debe realizar un seguimiento a lo largo de la ejecución del mismo para corregir las desviaciones sobre el plan, tomando las medidas oportunas.

Los tipos de seguimiento que se realizan son los siguientes:

- Estático: basado en el esfuerzo dedicado a una tarea
- Dinámico: basado en el avance de los productos obtenidos
- Predictivo: anticipa problemas futuros en función de variables actuales del proyecto, como la productividad.

Los tipos de seguimiento no son excluyentes, sino complementarios. En función de la información obtenida, se debe proceder a la replanificación del proyecto y a la toma de las acciones correctoras que se consideren necesarias.

Por último, una vez finalizado el proyecto se debe realizar cierre del mismo. Se liberan los recursos asignados, se guarda la información obtenida en un histórico de proyectos que ayude a la planificación de proyectos futuros, y se realiza un proceso de revisión para identificar las áreas de mejora y actualizar los estándares de la organización para futuros desarrollos.

En desarrollos importantes, la adecuada definición, gestión y seguimiento del proyecto es un factor crítico de éxito. Tanto es así que las técnicas que miden la madurez de los procesos de desarrollo, como CMMI o ITIL, le dan una importancia especial.

Las metodologías modernas de desarrollo la incluyen sin excepción (Métrica 3, SSADM, RUP...) y consideran que es una actividad de soporte que se lleva a cabo durante todo el ciclo de desarrollo.

Por último, será muy útil elaborar un balance final del proyecto, en el que se efectúe el resumen a alto nivel de los informes parciales de progreso del proyecto.

Este balance proporcionará siempre una valoración del cumplimiento de la definición general del proyecto, y deberá incluir cualesquiera otros aspectos que se estimen de interés, como pueden ser las principales incidencias o un resumen de la gestión de riesgos.

7. Conclusión.

La **Ley de Parkinson**, enunciada por el británico Cyril Northcote Parkinson en 1957, afirma que "el trabajo se expande hasta llenar el tiempo disponible para que se termine"; es decir: el tiempo asignado a un proyecto viene a ser exactamente el que efectivamente tarda en desarrollarse: si el tiempo es escaso el desarrollo será peor y se omitirán partes fundamentales.

7.1. Relación con el sistema educativo

- GS – DAW - DAM –Entornos de Desarrollo (PES)
- GS – ASIR – Implantación de Aplicaciones Web (PES)
- GS – DAM – Sistemas de Gestión Empresarial (PES)

8. Bibliografía

- Pressman, R. S. **Ingeniería del Software. Un enfoque práctico**, 3^a edición. Ed. McGraw-Hill, 2000.
- Sommerville, I.: **Ingeniería de Software**. 6^a Edición. Addison-Wesley Iberoamericana, 2002.
- Piattini, M. y otros: **Análisis y diseño detallado de Aplicaciones Informáticas de Gestión**. RA-MA, 1996.
- Cerrada, J. A.: **Introducción a la Ingeniería del Software**. 1^a Edición de 2000. Editorial Centro de Estudios Ramón Areces, S.A.