

ASI 2009-2010

Práctica de Planificación

Se trata de planificar un proyecto informático consistente en el desarrollo de 6 aplicaciones o módulos independientes, pero cuyo orden de realización se ha determinado teniendo en cuenta condicionantes temporales y restricciones en la disposición y despliegue de tecnología.

Para planificar el desarrollo de las aplicaciones se han previsto tres oficios_

- Análisis. Contempla el estudio de requerimientos y la determinación de los casos de uso internos.
- Diseño. Consiste en la ampliación detallada de los casos de uso y en el desarrollo de la vista de interacción y de la Vista estática. También se consideran actividades de diseño la elaboración de la Vista de la Máquina de Estados, la Vista de Gestión y la Vista Física.
- Programación. Consiste en la preparación del pseudocode de los métodos, la generación de código y el desarrollo de las pruebas de módulo y conjunto.

Los recursos disponibles son los siguientes:

- 2 Ingenieros encargados del análisis.
- 2 ingenieros encargados del diseño.
- 4 equipos de 2 programadores cada uno, encargados de la codificación y puesta a punto de programas.

Cada oficio se divide en dos fases.

La primera fase del Análisis abarca el estudio preliminar de los requerimientos y la estructuración general de casos de uso externos. La segunda fase del Análisis contempla la ampliación de los casos de uso y el estudio general de las colaboraciones y de la estructura de las clases colaboradoras.

En la primera fase del Diseño se terminan de detallar los casos de uso, tanto internos como externos, y se elaboran los diagramas de secuencia de la Vista de Interacción. En la segunda fase del Diseño se refinan los diagramas de secuencia, se elaboran los diagramas de colaboración y se crea el modelo definitivo de objetos, estructurando las clases resultantes y orientando sus propiedades fundamentales. También se trabaja la Vista de la Máquina de Estados y la de Actividades, así como se boceta la Vista de gestión y el despliegue físico.

En la primera fase de la Programación se diseña la lógica detallada de los programas y se codifican y prueban los módulos principales. En la segunda fase de la Programación se codifican el resto de los módulos y se ejecutan las pruebas de conjunto.

Las dependencias entre las actividades se organizan teniendo en cuenta que:

- La primera fase del Análisis es previa y condiciona la primera fase del Diseño, y supone el 60% del tiempo total de análisis.
- La segunda fase del Análisis condiciona a la segunda fase del Diseño, puede coincidir en el tiempo con la primera del Diseño, y consume el 40% restante del tiempo total de Análisis.
- La primera fase del Diseño condiciona a la primera fase de la Programación y supone un 60% del tiempo total de diseño.
- La segunda fase del Diseño se puede comenzar una vez terminada la primera y siempre que también se haya concluido la segunda fase del Análisis. Supone el 40% del tiempo de Diseño.
- La primera fase de la Programación depende de la disposición de los resultados de la primera fase del Diseño.
- La segunda fase de la Programación, precisa que se haya terminado la segunda fase del Diseño y que estén disponibles los programadores, que son los mismos que trabajan en la primera fase.
- Terminada la primera fase del Análisis, el ingeniero encargado pasa a desarrollar la segunda fase, y terminada esta la primera de la siguiente aplicación.
- Terminada la primera fase del Diseño, el ingeniero encargado, si no ha terminado la segunda fase de Análisis, pasa a apoyar a la codificación hasta que el análisis esté disponible, momento en el que iniciaría la segunda fase del Diseño. Esta ociosidad ha de intentar gestionarse con actividades de avance y retardo, pero las dependencias derivadas de la reasignación de recursos del diseño a la programación no se han de explicitar en la red.
- Terminada la segunda fase del Diseño, y si aún no está disponible la primera fase de análisis de la aplicación siguiente, el ingeniero diseñador pasa a apoyar a la codificación en las pruebas de conjunto, apoyo que se suspenderá en el momento en que disponga de análisis para iniciar el diseño de la siguiente aplicación. De nuevo esta ociosidad ha de intentar gestionarse con actividades de avance y retardo, pero las dependencias derivadas de la reasignación de recursos del diseño a la programación no se han de explicitar en la red.
- Los equipos de programadores se cruzan entre aplicaciones para compensar la incertidumbre de las estimaciones de tiempos. Este es el motivo por el que se duplican los recursos de programación. El cruce se efectúa de forma que el equipo de programación que está trabajando en la i -ésima aplicación, cuando termine pasará a trabajar en $(i+2)$ -ésima aplicación, lo que generará grandes holguras.

SE PIDE:

- Cumplimentar una tabla de dependencias cuyas columnas serán: Actividades predecesoras, Actividad en cuestión, Actividades sucesoras.
- Verificar, dibujando la escalera de dependencias, que el orden planteado entre oficios y fases es el correcto.
- Diseñar el patrón de planificación (se recomienda hacerlo en la tercera aplicación) empleando una red PERT y resolver las dependencias no deseadas.
- Dibujar la red PERT para la totalidad de las actividades y aplicaciones del proyecto, teniendo en cuenta reasignación implícita de equipos y/o recursos.
- Hacer una estimación de las duraciones y calcular tiempos y holguras de actividad y suceso.