

Solución Examen Parcial, Ingeniería del Software I.

21 de Noviembre de 2005.

NOMBRE:
GRUPO:

Nota: Contesta todas las preguntas en el enunciado, y entrégalo al terminar.

Ejercicio 1 (2.5 puntos)

Selecciona las respuestas correctas (puede haber 0 ó más). Cada respuesta incorrecta resta 0.1 puntos. Cada respuesta correcta suma 0.25 puntos. Una pregunta es correcta si todas las selecciones son correctas, de otra manera es incorrecta.

1. La estimación con puntos de función:
 - a. Es una técnica de estimación empírica, al igual que PERT.
 - b. **Es independiente de la experiencia de los programadores del proyecto.**
 - c. **Es independiente del lenguaje de implementación.**
 - d. Es adecuada para todo tipo de aplicaciones.
2. La estimación con COCOMO:
 - a. **Es una técnica de estimación empírica, al igual que Putnam.**
 - b. Es independiente del lenguaje de implementación.
 - c. Es independiente de la experiencia de los programadores del proyecto.
 - d. **Puede combinarse con técnicas de descomposición.**
3. Los equipos con estructura descentralizada democrática:
 - a. Suelen producir soluciones más rápidamente que los equipos con estructura descentralizada controlada.
 - b. **Suelen ser buenos cuando el software es complejo, y con poca modularidad.**
 - c. **No tienen un jefe permanente, sino que este va rotando.**
 - d. La estructura de gestión es plana, como en el descentralizado controlado.
4. Las causas de la llamada “crisis del software” fueron, entre otras:
 - a. Era frecuente que los plazos de entrega de los proyectos no se cumplieran.
 - b. Los programas fallaban a menudo.
 - c. **Había una mayor demanda de software.**
 - d. **El hardware era cada vez más potente.**
 - e. La productividad de los desarrolladores era baja.
5. El informe de viabilidad:
 - a. Tiene como objetivo principal informar sobre si el proyecto se realiza o no.
 - b. **Puede incluir una valoración de distintas alternativas.**
 - c. **Es la entrada a la planificación.**
 - d. Valora con más peso los aspectos económicos que los técnicos, ya que los primeros son más importantes de cara a la decisión de continuar con el proyecto.
6. Las diferencias entre una metodología y un modelo de ciclo de vida (MCV) son:
 - a. Que los modelos de ciclo de vida dan más información que las metodologías.

- b. **Que las metodologías dan información sobre cómo hacer las actividades (y los MCV no).**
 - c. Que las metodologías dicen qué productos intermedios hay que generar (y los MCV no).
 - d. Que hay metodologías – como el modelo de procesos del IEEE – que permiten la elección del MCV más adecuado.
 - e. **Que hay metodologías – como la metodología Métrica – que detalla cuál es el perfil de los participantes en cada actividad, incluidas las integrales.**
7. Las diferencias entre las maquetas y los prototipos son:
- a. Que las maquetas evolucionan hasta llegar al sistema final.
 - b. Que los prototipos son en realidad una técnica de análisis.
 - c. **Que las maquetas se “desechan” al comenzar el desarrollo.**
8. Las actividades integrales, como las que se muestran en el modelo de procesos del IEEE:
- a. **Incluyen por ejemplo las actividades de gestión de configuraciones.**
 - b. Se realizan al final del desarrollo, pero antes de la entrega.
 - c. Se realizan sólo durante el desarrollo, pero no en mantenimiento.
 - d. Incluye por ejemplo las actividades de operación y soporte.
9. El Work Breakdown Structure (WBS):
- a. Es un diagrama que se utiliza para estudiar distintas alternativas, y plasmarlas en forma de árbol.
 - b. Se suele utilizar junto con otras técnicas de estimación, por ejemplo las de descomposición.
 - c. Se suele utilizar junto con otras técnicas de toma de decisión, como los árboles de decisión.
 - d. **Identifica las tareas, que luego podremos incluir en diagramas de PERT.**
10. Los fundamentos de la Ingeniería del Software incluyen:
- a. La documentación.
 - b. Los métodos, que nos dicen el orden en el que se aplican los procedimientos.
 - c. Los procedimientos, que generalmente están soportados por herramientas, no como los métodos.
 - d. Las representaciones, como los diagramas de PERT y Gantt.
 - e. **Las herramientas.**

Ejercicio 2 (4.5 puntos)

La empresa TIKOTIKET te encarga la construcción de un software para la venta de entradas para acontecimientos deportivos y culturales. Se quiere que la aplicación funcione en la web, de tal manera que los usuarios accedan a través de un navegador. Los usuarios podrán comprar entradas de manera interactiva en tres pasos. En la primera pantalla introducen las características relativas al evento. En la segunda, introducen sus datos personales. Finalmente, en la tercera dan sus datos bancarios. Los usuarios podrán realizar consultas para obtener los eventos por tipo, por fecha y por localidad. Una vez obtenido el evento, se podrá realizar otra consulta sobre la disponibilidad y precio de las entradas. La disponibilidad de entradas se le proporcionará a la aplicación en un fichero externo.

Una vez realizada la compra, la aplicación permitirá imprimir las entradas en formato electrónico si los datos bancarios son válidos. En caso contrario se debe dar un mensaje de error. La aplicación debe pues comunicarse con un sistema de gestión de medios de pago, a la que mandará la información de la tarjeta del cliente, y de la que recibe si se autoriza el pago. Esta

comunicación, así como la introducción de datos bancarios debe realizarse de manera segura. La aplicación mantendrá una base de datos con la información de los clientes, que por seguridad ha de guardarse de manera encriptada. EL cifrado es un proceso matemático y lógico que se considera complejo. Se esperan periodos punta diarios de uso de la aplicación (suele haber más de un evento por día), típicamente entre las 18 y las 23 horas.

Además, la aplicación debe tener un módulo de mantenimiento, que será utilizado por los encargados de la empresa. Estos dispondrán de la posibilidad de realizar transacciones de alta y baja los distintos eventos de manera interactiva (esta actualización debe poder verse inmediatamente desde el módulo de los clientes), podrán obtener un listado con los clientes de cierto evento, así como un gráfico con estadísticas. Este módulo sólo podrá utilizarse desde equipos (PCs) dentro de la empresa TIKOTIKET, que tienen todos windows XP.

El gestor de TIKOTIKET parece no tener claro el estilo de la interfaz de usuario, tanto de la aplicación cliente como del módulo de mantenimiento. No obstante te pide que la aplicación tenga las características de usabilidad de las aplicaciones típicas de windows (uso de ratón, y presencia de menús, ventanas, scrolling, selección mediante cursor, teclas pre asignadas y ayuda on-line), así como minimizar el número de interacciones con la interfaz necesarias para adquirir las entradas. Has considerado la realización de la aplicación usando tecnología orientada a objetos (Java), y el gestor de TIKOTIKET te dice que quiere una versión básica de la aplicación lo antes posible para evaluarla.

Se pide:

- a) Elige razonadamente el modelo de ciclo de vida, y el modelo de desarrollo más adecuado (**0.5 puntos**).

Como el gestor de TIKOTIKET no tiene claro el estilo de la interfaz, habría que realizar una maqueta. Además, se prevee hacer la aplicación en incrementos, con una primera versión básica. Por tanto se hace necesario un modelo iterativo e incremental.

- b) Calcula los Puntos de Función ajustados, suponiendo media la complejidad de los parámetros, excepto las salidas, que se consideran de complejidad alta. Calcula el coste y el tamaño de la aplicación si los ratios de tu empresa señalan un coste de 700€ por punto de función y de 120 LDC por Punto de Función. (**3.5 puntos**).

Parámetros (todas de complejidad media, excepto las salidas, de complejidad alta).

Entradas: 3 (compra de entradas: eventos+datos personales+bancarios).
1 (autorización de pago del sistema de medios de pago).
2 (transacciones de alta y baja de eventos).
--

6 x 4 = 24

Salidas: 2 (impresión de la entrada+error dat. bancarios no válidos).
1 (información de la tarjeta al sistema medios pago).
2 (listado de clientes + gráfico con estadísticas)
--

5 x 7 = 35

Consultas: 4 (eventos por tipo+fecha+localidad+disponibilidad y precio)
--

4 x 4 = 16

Ficheros Internos: 1 (clientes)
1 (eventos)
--

2 x 10 = 20

Ficheros Externos: 1 (disponibilidad de entradas)
--

1 x 7 = 7

Total = 24 + 35 + 16 + 20 + 7 = 102 P.F. sin ajustar

Cálculo de las características generales del sistema:

1.	Comunicación de datos (datos del cliente por Internet)	: 4
2.	Funciones distribuidas (datos que el usuario procesa en local)	: 1
3.	Rendimiento	: 0
4.	Configuraciones fuertemente utilizadas (restricciones de seguridad)	: 2
5.	Frecuencia de transacciones (periodo punta diario)	: 3
6.	Entrada on-line de datos (todas las entradas son interactivas)	: 5
7.	Diseño para la eficiencia del usuario final (6 o más funciones, minimizar interacción para compra)	: 4
8.	Actualización on-line (actualización del fichero de eventos)	: 1
9.	Procesos complejos (matemáticos y lógicos: encriptación)	: 2
10.	Reusabilidad	: 0
11.	Facilidad de instalación	: 0
12.	Facilidad de operación	: 0
13.	Instalación de múltiples sitios (entornos idénticos en la empresa)	: 1
14.	Facilidad de cambio	: 0
Total		: 23

$$PF \text{ Ajustados} = 102 \times (0.01 \times 23 + 0.65) = 102 \times 0.88 = 89.76 \cong 90 \text{ PF}$$

$$\text{Coste} = 90 \times 700 \text{ €} = 63.000 \text{ €}$$

$$\text{Tamaño} = 90 \times 120 \text{ LDC} = 10.800 \text{ LDC} \cong 11.000 \text{ LDC}$$

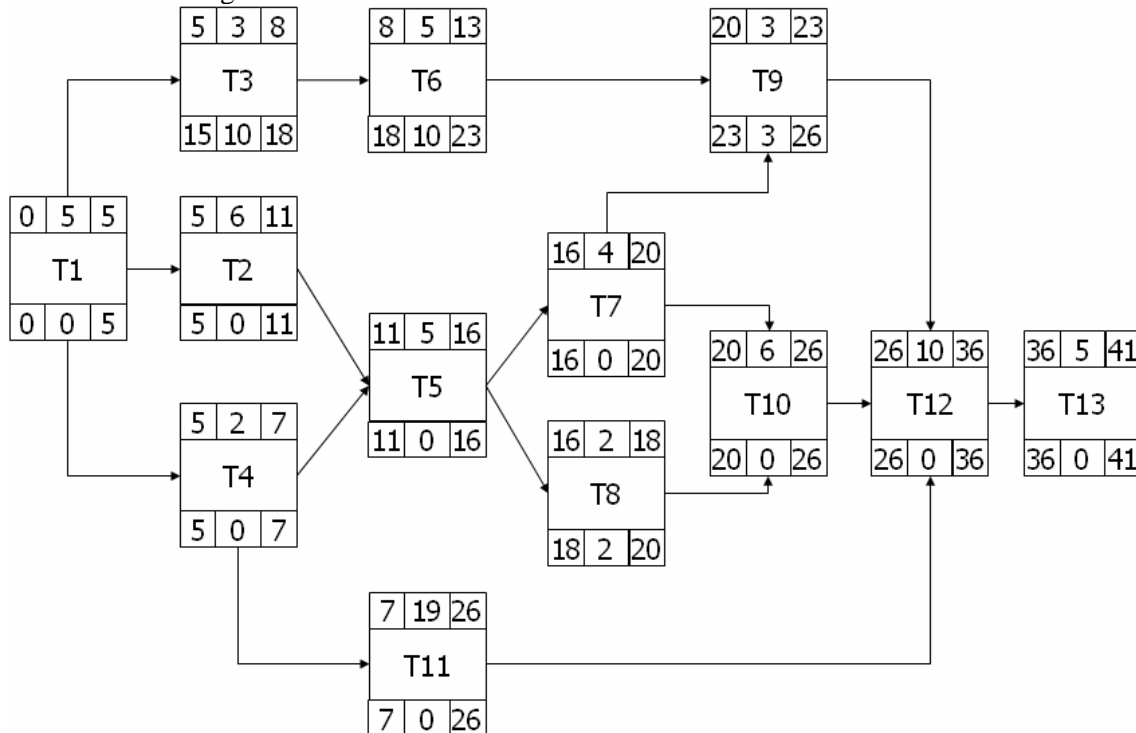
- c) Utilizando COCOMO básico, haz una estimación del tiempo total del proyecto y del esfuerzo de codificación. **(0.5 puntos)**

$$E = 2.4 \times 11^{1.05} = 29.76 \cong 30 \text{ p-m}, \text{ de los cuales, aproximadamente el 15\% son para la fase de codificación, es decir 4.5 p-m}$$

$$D = 2.5 \times 30^{0.38} = 9,1 \text{ meses}$$

Ejercicio 3 (3 puntos)

Dada la siguiente red de tareas.

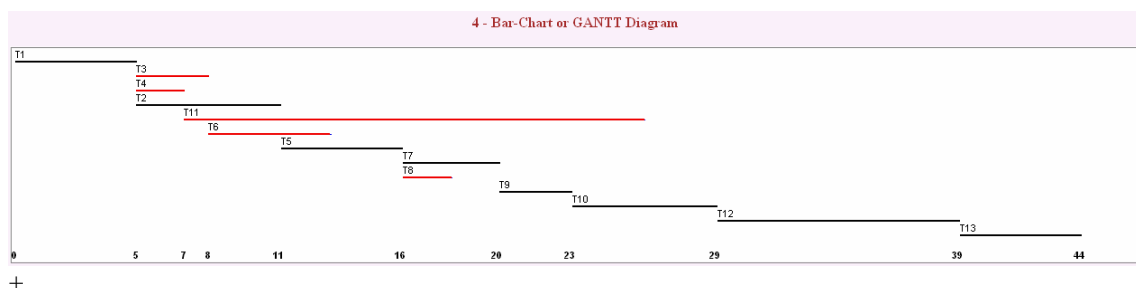


Se pide:

- El tiempo más temprano de inicio y fin de cada tarea (contesta sobre el diagrama).
- El tiempo más tardío de inicio y fin de cada tarea (contesta sobre el diagrama).
- La holgura de cada tarea (contesta sobre el diagrama).
- ¿Cuántos caminos críticos hay? ¿Cuáles son? (da la secuencia de tareas).

Hay 3 caminos críticos:
 T1-T2-T5-T7-T10-T12-T13
 T1-T4-T5-T7-T10-T12-T13
 T1-T11-T12-T13

- Un diagrama de Gantt equivalente (contesta en la página siguiente).



- ¿Puede tener ciclos un diagrama de PERT? Si como jefe de proyecto, eliges un modelo de ciclo de vida iterativo, ¿Cómo se modelan las iteraciones en un diagrama de PERT? (responde en 10 líneas o menos).

No pueden tener ciclos, ya que no podrían cumplirse las precondiciones para que las tareas pertenecientes a un ciclo se puedan realizar.
 Se modelan como tareas distintas.