

Examen Ingeniería del Software II.

1 de Septiembre 2004.

NOTA: Contesta en hojas separadas a cada pregunta.

Ejercicio 1 (3 puntos)

Contesta brevemente (tienes dos caras como máximo):

a) Explicar cómo afecta la reutilización a la fiabilidad y usabilidad del software.

Afecta positivamente a la fiabilidad puesto que si un componente ya probado se usa adecuadamente se incrementa la fiabilidad del sistema. Si además se utiliza repetidamente aumenta la confianza en el correcto funcionamiento de dicho componente. Pero existe el riesgo de que un componente no se use bien o se use fuera del alcance del componente.

Con respecto a la usabilidad, un efecto positivo es que para un componente reusable se pone más esfuerzo en su desarrollo. Sin embargo lo que en un entorno es aceptable en otro puede no serlo, lo que limita la reutilización del componente.

b) ¿Qué diferencias hay entre las pruebas de caja blanca y de caja negra? ¿Se pueden aplicar indistintamente en cualquier fase de las pruebas?

CAJA NEGRA	CAJA BLANCA
Se derivan de la especificación	Se derivan de la estructura del software y la implementación
Se aplica a unidades grandes y pequeñas de software	Se aplica a unidades pequeñas de software
El objetivo es comprobar que se satisfacen los requisitos funcionales	El objetivo es comprobar que el software funciona correctamente

c) ¿Qué diferencias y relaciones existen entre el mantenimiento preventivo y la reingeniería?

El objetivo de ambas es mejorar la mantenibilidad. Para realizar mantenimiento preventivo se hace reingeniería para establecer los modelos de diseño que se quieren mejorar como parte de ese mantenimiento preventivo.

Pero el mantenimiento preventivo es una labor más amplia que la reingeniería ya que trata en general de prevenir errores. Puede incluir la revisión periódica de equipos, periféricos y protocolos, la realización de pruebas y revisiones periódicas del software o el aumento del tamaño previsto de los ficheros o bases de datos.

d) Explica la diferencia entre la calidad del proceso y la calidad del producto y cómo se ven afectadas la una por la otra.

La calidad del producto se refiere a la calidad del software desarrollado (fiabilidad, rendimiento, usabilidad, eficiencia,...), mientras que la calidad del proceso se refiere a cómo es el proceso

mediante el cual se ha desarrollado ese software (gestionado, revisado de acuerdo a un plan de calidad, organizado,...). Un buen proceso no garantiza un buen producto, pero es difícil un buen producto con un mal proceso.

e) Comenta la diferencia entre entorno CASE de fuente única y entorno de apoyo integrado.

Un entorno de fuente única integra varias herramientas de un único vendedor. Es un modelo eficiente, pero es difícil añadir otras herramientas de vendedores distintos que quizá cubran otras etapas del desarrollo o que sean mejores para alguna etapa en particular.

En el entorno integrado se siguen estándares para que las herramientas se entiendan entre si, sin necesidad de que se hayan desarrollado por el mismo fabricante.

f) ¿Cuál es la diferencia entre la gestión del proceso y del proyecto?

La gestión del proceso consiste en definir las actividades estructurales y protectoras que se llevarán a cabo durante el desarrollo y adaptar las tareas estructurales a las características del proyecto. La gestión del proyecto implica planificar, supervisar y controlar esas actividades definidas.

Ejercicio 2 (1,5 puntos)

Dadas las siguientes situaciones, responder a las preguntas que se formulan:

- a) Se va a desarrollar la aplicación AMEC que depende de un sistema de gestión de datos construido por otra empresa. El sistema de gestión de datos está diseñado, pero aún no se ha comenzado a implementar. Se presupone que cuando AMEC vaya a ser integrado con el sistema de gestión de datos, éste estará operativo. ¿Qué riesgo(s) identificas? ¿En qué etapa del desarrollo? ¿Cómo reducirías dicho(s) riesgo(s)? ¿Cómo reducirías su impacto?

Un riesgo es que el sistema de gestión de datos no esté implementado cuando haya que integrarlo con AMEC.

El riesgo se puede materializar en la etapa de pruebas de integración. Para reducir el riesgo se pueden fijar reuniones para conocer exactamente el estado en el que está el sistema de gestión de datos, pero no se tiene control sobre él, puesto que está siendo desarrollado por otra empresa.

Para reducir el impacto del riesgo se debe encapsular la parte de ACME que deba conectarse con el sistema de gestión de datos.

- b) La forma de desarrollar software de la empresa para la que trabajas se caracteriza por lo siguiente: se planifican a ojo las tareas a realizar y los recursos necesarios; no se hace un seguimiento de los recursos consumidos, ni de los hitos alcanzados; aunque hay definido un proceso, no se comprueba si se sigue adecuadamente; la documentación es escasa y no siempre se elabora. ¿En qué nivel de madurez se puede situar esta empresa? ¿Qué acciones tomarías para que se incremente dicho nivel?

En el inicial.

Para pasar al nivel definido es necesario incluir:

- Procesos de gestión que controlen el proceso de desarrollo.
 - Calendario de desarrollo, planes, seguimiento, estimaciones.
 - Control de configuraciones, detección de defectos, inspecciones, auditorías.
- c) Una empresa de fabricación de envases dispone de una aplicación que gestiona los pedidos que recibe. La dirección de la empresa desea que la aplicación además de gestionar los pedidos, planifique, según la fecha de entrega del pedido, cuándo se deben empezar a fabricar los envases de un pedido. El problema es que solo se dispone del código fuente. ¿Qué tipo de mantenimiento se está solicitando? ¿Qué harías para facilitar el mantenimiento solicitado?

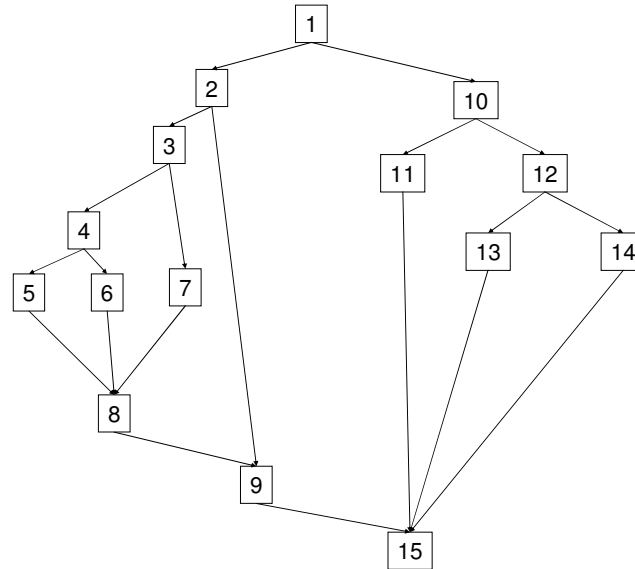
Mantenimiento perfectivo. Se puede facilitar mediante ingeniería inversa para obtener el modelo de diseño.

Ejercicio 3 (3 puntos)

Dado el siguiente fragmento de código:

```
Begin
  Lee DatosVisitante
  If Edad <> '' and NombreVisitante <> '' and DNIVisitante <> ''
  then
    If Edad >=18 then
     CodigoBarras=CodigoBarras + 1
      If Visitante = 'VIP' then
        If Cargo = 'Director' then
          ColorTarjeta = 'Dorado'
        Else
          ColorTarjeta = 'Rojo'
        Endif
      Else
        ColorTarjeta = 'Blanco'
      Endif
      PrintTARJETA CodigoBarras, NombreVisitante, Edad, DNIVisitante,
        ColorTarjeta
    Else
      Print 'Los menores de 18 años no pueden asistir a la exposición'
    Endif
  Else
    If Edad = '' then
      Print 'Debe introducir una edad'
    Else
      If NombreVisitante = '' then
        Print 'Debe introducir un nombre de visitante'
      Else
        Print 'Debe introducir DNI visitante'
      Endif
    Endif
  Endif
Endif
```

- a) Dibujar el grafo de flujo y calcular la complejidad ciclomática. ¿Qué indica el valor obtenido? (1 punto)



$$V(G) = 20 - 15 + 2 = 7$$

Indica el nº mínimo de casos de prueba necesarios para obtener cobertura de sentencias.

b) Generar los casos de prueba por la técnica del camino básico. (1 punto)

CP1: 1-2-3-4-5-8-9-15

CP2: 1-2-3-4-6-8-9-15

CP3: 1-2-3-7-8-9-15

CP4: 1-2-9-15

CP5: 1-10-11-15

CP6: 1-10-12-13-15

CP7: 1-10-12-14-15

CP1: Edad = 18

NombreVisitante = Juan

DNIVisitante = 124389

Visitante = VIP

Cargo = Director

Salida:

ColorTarjeta = Dorado

Mensaje: Tarjeta...

Incrementa 1 CodigoBarras

CP2: Edad = 18

NombreVisitante = Juan

DNIVisitante = 124389

Visitante = VIP

Cargo = Jefe de sección

Salida:

ColorTarjeta = Rojo

Mensaje: Tarjeta...

Incrementa 1 CodigoBarras

CP3: Edad = 18

NombreVisitante = Juan

DNIVisitante = 124389

Visitante ≠ VIP

Cargo = Director (**)

Salida:

ColorTarjeta = Blanco

Mensaje: Tarjeta...

Incrementa 1 CodigoBarras

CP4: Edad = 14

NombreVisitante = Juan

DNIVisitante = 124389

Visitante = VIP (**)

Cargo = Jefe de sección (**)

Salida:

ColorTarjeta = --

Mensaje: Los menores de...

CodigoBarras no se modifica

CP5: Edad = ''

NombreVisitante = Juan (*)

DNIVisitante = 124389 (*)

Visitante = VIP (**)

Cargo = Director (**)

Salida:

CP6: Edad = 18

NombreVisitante = ''

DNIVisitante = 124389

Visitante = VIP (*)

Cargo = Jefe de sección (*)

Salida:

```

ColorTarjeta = --
Mensaje: Debe introducir una edad
CodigoBarras no se modifica

```

```

ColorTarjeta = --
Mensaje: Debe introducir un n...
CodigoBarras no se modifica

```

```

CP7: Edad = 52 (*)
NombreVisitante = Juan (*)
DNIVisitante = ''
Visitante = VIP (**)
Cargo = Director (**)
Salida:
ColorTarjeta = --
Mensaje: Debe introducir DNI...
CodigoBarras no se modifica

```

(*) Da igual el valor que tenga este atributo, solo debe ser distinto de la cadena vacía.

(**) Da igual el valor que tenga este atributo.

c) Generar los casos de prueba para obtener cobertura de decisión. (1 punto)

```

D1: Edad <> '' and NombreVisitante <> '' and DNIVisitante <> ''
D2: Edad >=18
D3: Visitante = 'VIP'
D4: Cargo = 'Director'
D5: Edad = ''
D6: NombreVisitante = ''

```

Análisis de las decisiones:

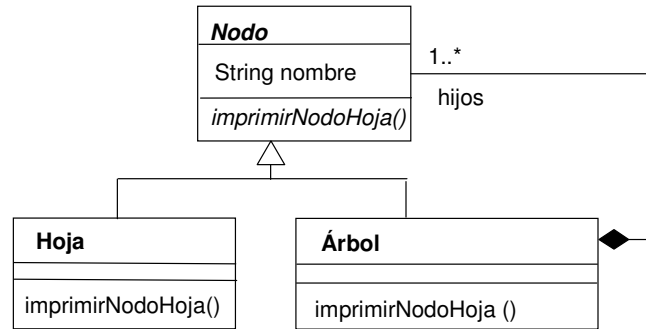
D1	D2	D3	D4	D5	D6
V	V	V	V	-	-
V	V	V	F	-	-
V	V	F	-	-	-
V	F	-	-	-	-
F	-	-	-	V	V
F	-	-	-	V	F
F	-	-	-	F	-

Los casos de prueba se definen de manera similar al apartado anterior, pero teniendo en cuenta los valores de las decisiones.

Ejercicio 4 (1,5 puntos)

Suponer una estructura de árbol. Cada nodo se identifica con un nombre y tiene n hijos. En el árbol se define la operación de imprimirNodoHoja () que imprime en pantalla el nombre de los nodos hoja.

- a) Utilizando patrones diseñar la estructura del árbol, indicando qué clases y métodos son abstractas y qué clases y métodos son concretos. (1 punto)



b) Definir las clases concretas. Puedes usar el lenguaje que prefieras. (0,5 puntos)

En la clase Hoja:

```

class Hoja extends Nodo{

    String obtenerNombre (){
        return this.nombre;
    }

    void imprimirNodoHoja (){
        println obtenerNombre();
    }
}
  
```

En la clase Árbol:

```

class Arbol extends Nodo{
    private Vector hijos;

    void imprimirNodoHoja (){
        Int i;

        for (i=0; size();){
            hijos.elementAt(i).imprimirNodoHoja ();
        }
    }
}
  
```

Nombre:

Grupo:

Ejercicio 5 (1 punto)

Contesta Verdadero o Falso a las siguientes preguntas, teniendo en cuenta que los errores descuentan 0,1 puntos (**contesta en esta hoja y entrégala junto con tu examen**):

1. La verificación de un producto consiste en comprobar que satisface las necesidades del cliente definidas en la especificación de requisitos. FALSO
2. Ante una petición de mantenimiento correctivo se analiza su impacto y se decide si se lleva a cabo. FALSO
3. Al realizar la actividad de auditoría en el ámbito de la gestión de la configuración del software, se comprueba si se ha informado de los cambios y se han actualizado los elementos de configuración relacionados. VERDADERO
4. La reutilización afecta siempre positivamente a la calidad del software. FALSO
5. En el plan de pruebas no se debe especificar el hardware y el software necesario para realizar las pruebas. FALSO
6. Cuando la toma de decisiones se realiza por consenso y la comunicación entre los miembros del equipo es horizontal se trata de una estructura basada en jefes de proyecto. FALSO
7. La reingeniería se hace cuando el mantenimiento es costoso y el software tiene un alto valor de negocio. VERDADERO
8. La gestión de configuración es un conjunto de actividades posteriores a la entrega del software al cliente, que completa el proceso de mantenimiento del software. FALSO
9. Las métricas aplicadas al proceso software mejoran por sí solas la calidad del software. FALSO
10. Para garantizar el éxito del uso de herramientas CASE en una empresa es necesario, entre otras cosas, haber establecido un modelo de proceso de desarrollo. VERDADERO