

[Página Principal](#) / [Mis cursos](#) / [IngSoft121](#) / [Parciales](#) / [Recuperatorio Parcial 2](#)

Comenzado el Tuesday, 16 de November de 2021, 10:57

Estado Finalizado

Finalizado en Tuesday, 16 de November de 2021, 12:52

Tiempo empleado 1 hora 54 minutos

Calificación Sin calificar aún

Pregunta 1

Finalizado

Puntúa como 1,00

¿Cuáles son las refactorizaciones más comunes?

Mencione al menos un ejemplo para cada una de ellas.

Las refactorizaciones mas comunes son:

- * Mejoras de métodos
- * Mejoras de clases
- *Mejoras de jerarquías

Ejemplo de mejora de métodos:

Agregar/eliminar parámetros:

- * Para simplificar las interfaces donde sea posible.
- * Agregar sólo si los parámetros existentes no proveen la información que se necesita.
- * Eliminar si los parámetros se agregaron originalmente "por las dudas" pero no se utilizan.

Ejemplo de mejoras de clases:

Desplazamiento de métodos:

- * Mover un método de una clase a otra.
- * Se realiza cuando el método actúa demasiado con los objetos de la otra clase.
- * Inicialmente puede ser conveniente dejar un método en la clase inicial que delegue al nuevo (debería tender a desaparecer).

Ejemplo de Mejoras de Jerarquías:

Reemplazar condicionales con polimorfismos:

- * Si el comportamiento depende de algún indicador de tipo, no se está explotando el poder de la OO.
- * Reemplazar tal análisis de casos a través de una jerarquía de clases apropiada.

Comentario:

Estos son explicaciones no ejemplos.

4

Pregunta **2**

Finalizado

Puntúa como 1,00

Explique exhaustivamente el proceso de desarrollo: Iterativo.

El proceso de Desarrollo Iterativo aborda el problema de "todo o nada" del modelo de cascada, combina beneficios del prototipado y del cascada.

Desarrolla y entrega el SW incrementalmente, cada incremento es completo en sí mismo.

Provee un marco para facilitar el testing (el testing de cada incremento es más fácil que el testing del sistema completo).

Puede verse como una "secuencia de cascadas".

El feedback de una iteración puede usarse en iteraciones futuras.

Ventajas:

Entregas regulares y rápidas.

Reduce riesgo.

Acepta cambios naturalmente.

Permite feedback del usuario.

Prioriza requisitos.

Desventajas y/u Debilidades:

Sobrecarga de planeamiento en cada iteración.

Posible incremento costo total (trabajo en una iteración puede deshacerse en otra).

Arq. y diseño pueden ser afectados con tantos cambios.

Uso:

Para empresas donde el tiempo es esencial.

Donde no puede enfrentarse el riesgo de proyectos largos.

Cuando los requerimientos son desconocidos y sólo se comprenderán con el tiempo.

Desarrollo iterativo: Enfoque modelo en espiral.

Cada ciclo de la espiral:

1. Identificar objetivos, distintas alternativas para conseguirlos, y restricciones.
2. Evaluar alternativas en base a objetivos y restricciones (considerar riesgos).

Desarrollar estrategias para resolver incertidumbre y riesgos.

Comentario:

Incompleto

8

Pregunta **3**

Finalizado

Puntúa como 1,00

¿Qué es el testing de caja negra? Enuncie al menos dos criterios dentro de esta categoría.

El testing de Caja negra es bueno para detectar errores de entrada/salida, es decir errores funcionales, los cuales son útiles a alto nivel, donde se busca

analizar el comportamiento funcional del sistema o partes de éste.

La especificación de la caja negra está dada para diseñar los casos de test, se utiliza el comportamiento esperado del sistema.

El comportamiento esperado está especificado, entonces sólo se definen test para el comportamiento esperado especificado.

Criterios:

- * Particionado por clase de equivalencia.
- * Análisis de valores límites
- * Grafo de causa-efecto

Comentario:

Esto no explica qué es el testing de caja negra solo menciona características de el...

5

Pregunta 4

Finalizado

Puntúa como 1,00

Considere el siguiente fragmento de código:

```
public static bool Contiene(string palabra, string cadena)
01. {
02.     for (int i=0; i<=palabra.Length-cadena.Length; i++)
03.         if (palabra[i]==cadena[0])
04.             {
05.                 bool contenida = true;
06.                 for (int j=1; j<cadena.Length; j++)
07.                     {
08.                         if (cadena[i+j] != cadena[j])
09.                             contenida = false;
10.                         if (contenida)
11.                             return true;
12.                     }
13.             }
14.     }
15.     return false;
16. }
```

a) Construya un conjunto de casos de tests para este código que cumpla con el criterios de cobertura de ramificaciones.

b) Construya el grafo de definición-uso etiquetando apropiadamente los: def., uso-c y uso-p.

c) ¿Porqué las preguntas a y b están en el orden dado?

a)

Cobertura de ramificacion:

Caminos = { (1,2,3,4,5,7,8), (1,9), (1,2,3,4,5,6,7,4) }

Tests = {(palabra="hola", cadena="ho", return=true), (palabra="a", cadena="b", return=false), ((palabra="acasa", cadena="caz"), return=IndexOutOfBounds)}

b) Resuelto en las imagenes.

c) Nos pide que primero hagamos los caminos con sus tests para que tratemos de entender el programa, luego de realizar el grafo se realizan los test para

ver si cumple con los criterios de cobertura de ramificacion

 [b.jpeg](#) [b.png](#)

Comentario:

b) Me imagino que el b.jpeg es para decir la relación sentencia estados, tomando esto como válido, está mal creer que el estado 1 es el for

Si damos el programa anotado es para que usen esos nombres para los estados.

en este caso está mal no nombrar la sentencia 12

al no entrar al primer if a esa flecha le falta el p-use

las flechas que salen del estado 4 deberían tener las variables j y cadena como p-use

faltan p-use

del estado 8 puedo entrar o no y luego de entrar o no vuelve al estado 4 eso tampoco está en el grafo

5

Pregunta **5**

Finalizado

Puntúa como 1,00

Describe exhaustivamente la Administración de riesgos.

Cualquier proyecto puede fallar debido a eventos no previstos.

La administración de riesgo es un intento de minimizar las chances de fallas y tiene como objetivo minimizar el impacto (en los costos, calidad y tiempos) de la materialización de los riesgos.

La administración de riesgos se lleva a cabo durante dos etapas:

Evaluación de Riesgos => Se realiza durante el planeamiento del proyecto.

Control de Riesgos => Se lleva a cabo durante la realización del proyecto.

Evaluación de riesgos:

*Identificación del riesgo:

Identifica los posibles riesgos del proyecto y la forma de hacerlo es mediante listas de control, experiencias pasadas, brainstorming, etc.

Algunos factores de riesgos mas importantes:

1. Personal: Entrenamiento insuficiente o inapropiado
2. Tiempos y costos irreales
3. Discrepancia con la interfaz de usuario
4. etc.

*Análisis de riesgo y definición de prioridades:

Como la cantidad de riesgos puede ser grande, se debe priorizar la atención en las areas de alto riesgo.

Para ello se establece la probabilidad de materialización de los riesgos identificados y la pérdida que éstos originarían,

Ordenar de acuerdo al "valor de exposición al riesgo" (RE):

$RE = \text{prob. ocurrencia indeseada} * \text{impacto ocur. indeseada}$

Control de riesgos:

Si es posible evitar los riesgos => evitarlos

En otros casos sino se puede evitar los riesgos, planear y ejecutar los pasos necesarios para mitigar los riesgos.

Ejemplo de mitigación de riesgos: "Demasiados cambios de requerimientos"

Comentario:

10

Promedio del parcial = 64%

Nota = 4,8

[◀ Recuperatorio Parcial 1](#)

Ir a...

[0 - Presentación ▶](#)