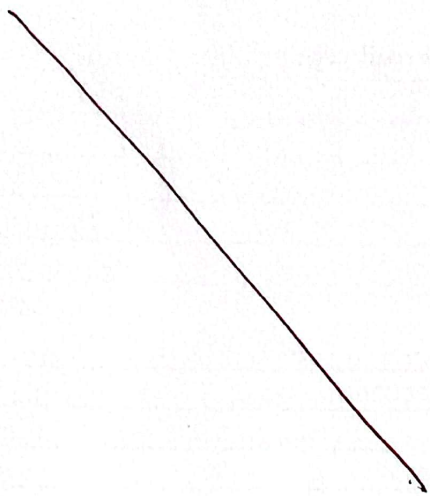


EXAMEN FINAL DE INGENIERÍA DEL SOFTWARE II
GRUPO: CCOMP6-1

1. Elabore un plan de mantenimiento para su proyecto de laboratorio. (3 pts.) (habilidad b, c, k)



2. ¿Por qué el mantenimiento de software es un proceso crítico dentro del ciclo de vida del desarrollo de software? (1 pts.) (habilidad f)

0,75
Mantiene el producto de software utilizable, además
se encarga de los posibles cambios o modificaciones
que el usuario pueda solicitar o algún problema
omitido en testing. Por esto, el mantenimiento es eterno.

- 2,75
3. Considerando las exposiciones finales de los papers, responda las siguientes preguntas. (7 pts.) (habilidad f)

0,5 ✓ 3.1 Las dos metodologías más utilizadas (según las exposiciones) son: Scrum y XP

3.2 ¿Qué se utiliza en la estimación de esfuerzo?

<input checked="" type="checkbox"/>	Tiempo
<input checked="" type="checkbox"/>	Recursos
<input checked="" type="checkbox"/>	Dinero
<input checked="" type="checkbox"/>	Asesoría
<input type="checkbox"/>	Todos es correcto

3.3 Los siguientes son tipos de equipos de desarrollo ágiles:

<input checked="" type="checkbox"/>	Control mixto
<input type="checkbox"/>	Descentralizado
<input checked="" type="checkbox"/>	Democrático
<input checked="" type="checkbox"/>	Centralizado
<input checked="" type="checkbox"/>	Todos son correctos

3.4 ¿por qué es importante la estimación?

<input checked="" type="checkbox"/>	Permite elaborar un presupuesto adecuado
<input checked="" type="checkbox"/>	Mejora la calidad del software
<input type="checkbox"/>	Permite gestionar las compensaciones
<input checked="" type="checkbox"/>	Permite medir el software
<input checked="" type="checkbox"/>	Todas son correctas

3.5 La exactitud el la medida de cuan cerca está un resultado a su valor real.

3.6 ¿Cuáles de los siguientes son pasos/características para construir un equipo eficiente?

<input checked="" type="checkbox"/>	Establecer metas comunes.
<input checked="" type="checkbox"/>	Definir roles.
<input checked="" type="checkbox"/>	Comunicación.
<input checked="" type="checkbox"/>	Compromiso.
<input checked="" type="checkbox"/>	Tener una estrategia.

3.7 Relacione según corresponda

A	Prácticas sociales	<input checked="" type="checkbox"/>	Diseño de software
B	Prácticas de gestión	<input checked="" type="checkbox"/>	Tablero Kanban
C	Prácticas de desarrollo	<input checked="" type="checkbox"/>	Crear repositorio de trabajo

3.8 El objetivo de la gestión de proyectos es:

<input type="checkbox"/>	Tiempo
<input type="checkbox"/>	Alcance
<input checked="" type="checkbox"/>	Costos
<input checked="" type="checkbox"/>	Calidad
<input checked="" type="checkbox"/>	Todo es correcto

3.9 La estimación es el proceso que permite predecir el esfuerzo para desarrollar un proyecto de software.

3.10 La técnica de puntos de casos de uso no considera la gestión de riesgos.

3.11 La gestión de riesgos se basa en:

0,25

<input checked="" type="checkbox"/>	Control
<input checked="" type="checkbox"/>	Evaluación
<input checked="" type="checkbox"/>	Prevención
<input checked="" type="checkbox"/>	Identificación
<input type="checkbox"/>	Eliminación

3.12 Un riesgo está compuesto de:

0,5

<input checked="" type="checkbox"/>	Mal funcionamiento de software
<input checked="" type="checkbox"/>	Gravedad
<input type="checkbox"/>	Código
<input type="checkbox"/>	Tipo
<input type="checkbox"/>	Todos son componentes

3.13 Un riesgo positivo es aquel que contribuye en algún aspecto al proyecto, pero puede afectar en otro

0 () VERDADERO (x) FALSO

3.14 Un riesgo positivo es aquel que afecta de manera muy elevada el desarrollo del proyecto.

0 (x) VERDADERO () FALSO

4. Explique qué es Refactoring y ejemplifique un par de técnicas (2 pts.) (habilidad i)

2 Refactoring pertenece al tema de mantenimiento. Y es mejorar, eliminar redundancias, ambigüedades en la implementación (código) de software. Para modificar la implementación se modifica el diseño, es por eso que tan bien se ve afectado

Ejemplos

(1) Algunas veces tenemos Clases padre que tienen muchos atributos pero solo lo hereda a una clase hijo.
Solución: Que solo la clase hijo tenga los atributos del padre.

(2) El mal uso de variables. En el punto donde se coloca una expresión demasiado larga ejemplo:

→ `Print(ball_node(32,8) + value.tree.root(-2))`

Si dicho valor es importante para los demás desarrolladores, entonces se debe almacenar en una variable que tenga un nombre muy descriptivo.

5. Seleccione un paper que Usted no haya expuesto y defina: ¿quiénes lo expusieron?, idea principal del paper (clara y detallada), tema del curso asociado al paper y apreciación de cómo expusieron el paper sus compañeros. (3 pts.) (habilidad i)

Expuesto por: Emmanuel Galdos

Idea Principal: No hay una buena metodología o forma sencilla para que personas principiantes puedan desarrollar software. El paper brinda MIDIS para el desarrollo individual de software.

Tema del Curso: Gestión de proyectos, gestión de equipos.

Apreciación: Buena exposición, sin embargo sentí que profundizó mucho en los conceptos y en el paper fue muy rápido.

6. Luego de evaluar técnicamente un sistema heredado, se obtienen 4 tipos de sistemas, esta evaluación debe determinar las acciones a seguir con esos sistemas; complete la siguiente tabla considerando las acciones a seguir según el tipo (2 pts.) (habilidad i)

BAJA CALIDAD Y BAJO VALOR	BAJA CALIDAD Y ALTO VALOR
ALTA CALIDAD Y BAJO VALOR	ALTA CALIDAD Y ALTO VALOR

7. De un ejemplo de rastreo de requisitos, utilizando artefactos de su proyecto de laboratorio (2 pts.) (habilidad b,c,k)

