



Carrera: Ingeniería en Informática.

Materia: Ingeniería del Software

Fecha: 30/10/2013

Se regulariza con un mínimo de 40 pts. y se promociona con un mínimo de 80 pts.

Consignas:

Ejercicio 1:

Marque con una cruz la respuesta correcta. Valor total 15 puntos, cada respuesta incorrecta resta 1,1 puntos.

1. En medición de software los *indicadores* se obtienen a partir del uso de métricas.
 - a. Verdadero. (OK)
 - b. Falso
2. ¿Cuáles de las siguientes *métricas* pueden utilizarse para evaluar el diseño a nivel de *componentes*?(+1)
 - a. número de parámetros de datos de entrada. (OK)
 - b. cantidad promedio de módulos externos invocados desde un módulo interno del sistema. (OK)
 - c. complejidad estructural entre componentes. (Rpta: esta es a nivel arquitectura, NO corresponde)
 - d. número de errores en la documentación.
 - e. tiempo de desarrollo promedio de un módulo del sistema.
3. Cuanto *mayor* sea el valor de la medida de una métrica que evalúe el nivel de acoplamiento entre componentes o clases de un modelo: (+1)
 - a. Disminuye la probabilidad que los cambios puedan tener efectos extensivos al resto del sistema.
 - b. Aumenta la complejidad del mantenimiento.(OK)
 - c. Disminuye el grado de reutilización. (OK)
 - d. Disminuye la complejidad del diseño de casos de prueba.
4. Los ‘casos de pruebas’ pueden generarse automáticamente.
 - a. Verdadero.
 - b. Falso. (OK)
5. La probabilidad de la operación libre de fallas de un software durante un lapso de tiempo específico se refiere a la: (1)
 - a. Usabilidad.
 - b. Fiabilidad. (OK)
 - c. Verificabilidad.
 - d. Tasa de corrección de defectos.
 - e. Evolución de defectos a lo largo del ciclo de vida.
6. Desde el punto de vista costos (tangibles e intangibles) las consideradas *fallas internas* (errores) son más costosas que las *fallas externas* (fallos).
 - a. Verdadero.
 - b. Falso. (OK).
7. Los sistemas heredados son uno de los retos del siglo XXI según Sommerville. Generalmente, tienen un papel crítico en la organización y dependen de tecnologías de hardware y software obsoletas. En relación a este tipo de sistemas se puede afirmar que: (+1)
 - a. Incluyen procesos y procedimientos, además de hardware y software. X
 - b. Condicionan las posibilidades de actualización de hardware. X



Universidad Católica de Santiago del Estero

Departamento Académico Rafaela

B. Hip. Irigoyen 1502 – Rafaela – Santa Fe – República Argentina

Te: 03492-432832-433408-433550 - ucsedar@ucse.edu.ar

Parcial N° 1

- c. No proporcionan servicios esenciales del negocio.
 - d. Sólo deben mantenerse por los datos que manejan.
 - e. Todas las anteriores.
 - a. Ninguna de las anteriores.
8. ISO9001 es una *norma* específica para procesos de desarrollo de software.
- a. Verdadero.
 - b. Falso. (X)
9. ¿Qué son los métodos de la ingeniería del software? (Sommerville)
- a. Programas de computadoras y documentación asociada
 - b. Un conjunto de actividades cuya meta es el desarrollo o evolución del software.
 - c. Enfoques estructurados para el desarrollo de software que incluyen modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos. (X – sommerville figura 1.1)
 - d. Una representación simplificada de un proceso de software, presentada desde una perspectiva específica.
10. ¿Qué factores han precipitado la generación de sistemas basados en computadores más complejos y sofisticados?
- a. Grandes incrementos en la capacidad de memoria y almacenamiento de las computadoras.
 - b. Gran variedad de opciones de entradas/salidas
 - c. Profundos cambios en las arquitecturas de computadoras.
 - d. Todas las anteriores. (X – Pressman 1.1, página 4)
11. La información acerca de la historia de un cambio se mantiene en:
- a. El documento de especificación del programa. X
 - b. El log de cambios al repositorio del proyecto.
 - c. El registro de solicitudes de cambio. X
 - d. El código afectado. X
 - e. Ninguno de los anteriores.
12. En Ingeniería de Software los Procesos:
- a. Brindan soporte a las actividades de producción de software.
 - b. Indican los métodos y herramientas a utilizar. (X)
 - c. Son los paradigmas que representan un enfoque particular para la construcción del software. (X)
 - d. Se ocupan de las teorías y los fundamentos de los sistemas de software.
 - e. Todas las anteriores.
 - f. Ninguna de las anteriores.
13. La relación entre ‘entregas’ y ‘versiones’ de un producto de software es constante: Cada ‘versión’ del producto implica una ‘entrega’
- a. Verdadero.
 - b. Falso. (X - Siempre existen más versiones de un sistema que las entregas.)
14. La diferencia entre las métricas de Puntos de Función y las métricas de Puntos de Función ampliadas (Puntos de Características y Puntos de Función 3D) reside en: (1)
- a. El peso asignado a las distintas complejidades que se analizan.
 - b. Los parámetros de medición. (X)
 - c. La fórmula de cálculo.
 - d. El tipo de de sistemas y aplicaciones a los que se aplican. (X)



Parcial N° 1

- e. Todas las anteriores.
15. ‘Calidad de Concordancia’, significa que: (1)
- Las especificaciones de CU se aplican durante el diseño.
 - Las especificaciones de diseño se aplican durante la fabricación. (X)
 - Las especificaciones de software de base concuerdan con la plataforma del cliente.
 - Ninguna de las anteriores.
16. En un proyecto se está utilizando una numeración de versiones que sigue la convención “Id 9.9.9.9”. (20 puntos)
- Esta se corresponde a identificar:
 - Release, Nivel, Versión, Instalación (X)
 - Versión, Nivel, Release, Instalación
 - Versión, Release, Nivel, Instalación

Rta: convención de nomenclatura:

• <Id> <9>.<9>.<9>.<9>

→ **Instalación:** se incrementa cuando se realiza una modificación debido al entorno donde se instalará.

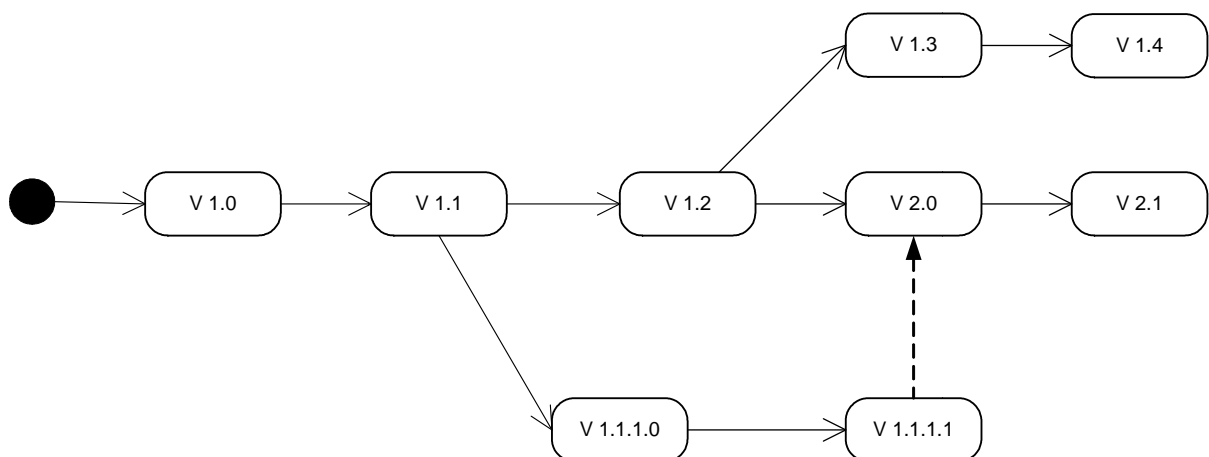
> **Versión:** se incrementa cuando se corrige un fallo o problema.

→ **Nivel:** se incrementa cuando se produce una modificación leve en el producto.

→ **Release:** se incrementa cuando se produce una modificación importante en el producto.

> Identificación del producto.

- Dado el siguiente gráfico (grafo) que representa la evolución de un producto de software, que utiliza la convención de referencia, indique que representa cada rama del mismo:



Rta: Tenemos la rama principal o “versión principal”, luego los branch (ramas) que representan variantes menores, en la rama V1.1.1.0 se origina por la corrección de un fallo o problema. La rama V 1.3 se origina cuando se realiza una leve modificación que se separa del tronco principal. Luego, tenemos la unión (merge) de un branch, al tronco principal o versión principal.



Parcial N° 1

17. ¿Por qué motivo podríamos querer poner bajo control de configuración el software utilizado para desarrollar nuestro producto, como el compilador, el framework de desarrollo, la herramienta de modelado de Bases de datos, etc.? ¿Es recomendable esta medida? (15 puntos)

Rpta: Muchas organizaciones también ponen los productos hardware y software utilizados durante el desarrollo bajo control de configuración. Como estos productos son necesarios para desarrollar el software, deberán estar disponibles cuando se realicen cambios sobre su configuración, principalmente cuando estamos hablando de software que se discontinúa o tiene variantes significativas en cada versión. Aunque no es normal que haya problemas, es posible que una nueva versión de una herramienta produzca resultados diferentes que la versión usada (un compilador, por ejemplo). Es **ABSOLUTAMENTE** recomendable en estos casos.

18. Un equipo de desarrollo de software quiere iniciarse en los ‘procesos de medición y mejora’, y uno de los aspectos que consideran importante es poder determinar el “Esfuerzo que les implica el mantenimiento de los productos de software”. Con este objetivo, deben definir un conjunto de ‘métricas e indicadores’. De acuerdo a este objetivo, determine: (20 puntos)
- Entidad y concepto a evaluar.
 - Modelo de conceptos y atributos relevantes.
 - Seleccione un atributo y especifique una métrica posible.
 - Para la métrica definida, explicita un indicador identificando su tipo.

Rta: Tenemos un **objetivo claro**: Esfuerzo que implica el mantenimiento de los productos de software. Es un modelo dirigido por “Metas”, los pasos son:

- Identificar lo que se quiere conocer o aprender.
- Identificar los **subobjetivos**.
- Identificar **entidades y atributos** relacionados con los objetivos secundarios.
- Formalizar los objetivos de la medición.
- Identificar **preguntas cuantificables** y los **indicadores** relacionados que se emplearán como apoyo para lograr los objetivos.
- Identificar elementos de datos que se recopilarán para construir los indicadores para responder las preguntas.
- Definir las medidas e identificar acciones para implementarlas medidas.
- Preparar un plan para implementar las medidas.

Entidades que se pueden medir Tiempo de desarrollo, Productividad, etc Atributos relacionados a esas entidades: líneas de código generadas.

Luego definir las métricas para estos atributos.

19. Es usted el jefe de un proyecto de desarrollo de un sistema de comunicaciones en el cual ha estimado tener 30 KLDC. El software a desarrollar es de alta complejidad pero afortunadamente podremos contar con personal de muy alta cualificación y experiencia específica en el tipo de software a desarrollar. Teniendo en cuenta que el costo del salario mensual promedio de cada persona es de 13500\$ mensuales:
- Calcule el esfuerzo de desarrollo de este sistema, tiempo, personal y costo teniendo en cuenta los valores dados (20 puntos).
 - Para el caso anterior, ¿sería más rentable en tiempo y dinero, emplear personas de nivel medio cuyo salario es de 11500 \$ mensuales? Justifique su respuesta (10 puntos).

Nota: En este proyecto podemos considerar que el valor del FAE es : 1.15 (complejidad) * 0.7 (calidad prog) * 0.91 (experiencia) = 0,73255.

$$E = \text{Esfuerzo} = a \text{ KLDC}^e * \text{FAE}$$

$$T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo}^d$$



Universidad Católica de Santiago del Estero
Departamento Académico Rafaela
B. Hip. Irigoyen 1502 – Rafaela – Santa Fe – República Argentina
Te: 03492-432832-433408-433550 - ucsedar@ucse.edu.ar

Parcial N° 1

P= Personal = E/T

PROYECTO SOFTWARE	a	e	c	d
Orgánico	3.2	1.05	2.5	0.38
Semi - acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.2	2.5	0.32

Rta: Parte a:

Esfuerzo = 83,35 personas/mes

Con este dato, hallamos el resto de variables:

Costo = $83,35 * 13.500 = 1.125.225$

Tiempo = $2,5 * 83,35^{0.38} = 13,42$ meses

N° medio de personas = $83,35/13,42 = 6.2$ personas -> 7 personas por mes

Parte b:

Al cambiar la experiencia del grupo, el proyecto cambia de ser un proyecto Orgánico a Semi-Acoplado, por lo tanto, los valores serían:

FAE es : 1.15 (complejidad) * 0.7 (calidad prog) * 1 (experiencia) = $0,805$.

Esfuerzo = $3,0 * 30^{1,12} * 0.805 = 108,966$ personas/mes

Costo = $108,996 * 11500 = 1.253.114$ -> Esto es MAS que en el caso anterior...

Tiempo = $2,5 * 108,966^{0.35} = 12,91$

N° Medio de Personas = $108,966 / 12,91 = 8,44$ -> 9 personas por mes

No es conveniente, ya que tengo que pagar más... me lleva más personas en casi el mismo tiempo, sabemos que coordinar más personas es siempre complejo.