

- Cada pregunta de la **parte múltiple opción** contestada correctamente tiene un valor de 2 puntos. Cada pregunta incorrecta de la múltiple opción resta 0,66 puntos.
- **El puntaje total del parcial es 50 puntos.**

## Múltiple Opción

- 1) La puesta en funcionamiento de un sistema complejo cuya parte principal es software:
  - a. puede realizarse mediante una estrategia "big-bang" que consiste en que a partir de cierto instante el sistema pase a estar en producción de forma completa
  - b. (a) o mediante una estrategia de implantación paulatina, la que consiste en la incorporación paulatina de distintas áreas geográficas al uso del sistema
  - c. tanto en una estrategia de implantación big-bang como en una paulatina, en caso de que se trate de un sistema nuevo que no sustituye a un sistema anterior se puede llevar a cabo un procesamiento en paralelo.
  - d. El procesamiento en paralelo, además de permitir detectar problemas en los procedimientos, resulta en general un mecanismo de verificación que permite lograr un excelente nivel de cobertura funcional con muy poco esfuerzo.

---

- 2) La empresa de software "XX" dedica el 89% de su esfuerzo a tareas de mantenimiento del producto de software "A" y 11% a tareas de desarrollo de software nuevo.  
A partir de esa información resulta posible concluir que:
  - a. El producto "A" presenta severos problemas de calidad.
  - b. El mantenimiento llevado a cabo sobre el producto "A" está centrado en dar solución lo antes posible a los problemas que surgen, sin tomar en cuenta la preservación de los atributos de calidad de "A" relacionados con la facilidad de mantenimiento.
  - c. La empresa "XX" enfrenta serios problemas de falta de eficiencia y de calidad.
  - d. Falta información para determinar si la empresa "XX" está frente a un problema o no en lo atinente al mantenimiento.

---

- 3) Los sistemas legados:
  - a. corresponden a sistemas desarrollados hace poco tiempo con tecnologías de última generación y que justamente por esa razón suelen presentar un nivel de madurez poco adecuado.
  - b. frecuentemente corresponden a sistemas que resultan vitales para el funcionamiento de las organizaciones y suelen presentar costos de mantenimiento bajos y decrecientes
  - c. entre las razones que explican la supervivencia de estos sistemas está lo difícil que suele resultar para una organización encarar la sustitución de los sistemas centrales que soportan su funcionamiento, tanto con una estrategia de sustitución big-bang como paulatina
  - d. se pueden rejuvenecer aplicando sucesivamente un proceso de ingeniería directa seguido de etapas de ingeniería reversa a partir del cual se obtiene un sistema funcionalmente equivalente al original.

---

- 4) La planificación de la liberación (puesta en explotación) de un software, conviene que comience:
  - a. una vez que la arquitectura está estable y fue validada respecto a los requerimientos no funcionales definidos
  - b. una vez que se dispone de una versión beta en condiciones de ser evaluada por usuarios y cliente, para permitir asegurar que el esfuerzo de planificación se pueda desarrollar sobre bases firmes
  - c. al inicio del proyecto, ya que es el único momento en el que corresponde llevar a cabo la planificación del proyecto
  - d. al inicio del proyecto, ya que actividades relacionadas con la liberación tales como documentación, entrenamiento, migración de información de un sistema anterior y soporte a la solución de problemas pueden tener un impacto significativo en el esfuerzo y plazo necesarios para abarcar el alcance del proyecto

---

- 5) De acuerdo con el estándar ISO/IEC 9126, la Comprensibilidad, Facilidad de estudio y Operabilidad del software determinan el nivel de cual de sus siguientes características:
  - a. Confiabilidad
  - b. Eficiencia
  - c. Usabilidad
  - d. Mantenibilidad

- 6) Entre los problemas que plantea el mantenimiento están:
- problemas del personal, por ejemplo:  
Comprensión limitada, cambios en las prioridades de la gerencia, problemas de motivación
  - Problemas Técnicos, por ejemplo:  
Especificaciones de diseño inadecuadas, Documentación no actualizada, Dificultades para identificar el impacto de un cambio y para armar ambientes de prueba adecuados
  - Necesidades conflictivas, por ejemplo:  
Solucionar lo antes posible un problema en un sistema que el personal de mantenimiento no domina y por otro lado mantener o mejorar los atributos de calidad del producto y entre ellos la facilidad de mantenimiento
  - costo: todos los problemas contribuyen a elevar los costos de mantenimiento
- Seleccionar la opción correcta:
- Sólo IV es correcta
  - Sólo I y II son correctas
  - Sólo I y III son correctas
  - Las opciones I, II, III y IV son correctas**
- 
- 7) La Madurez de una Organización se determina por:
- El desempeño del conjunto de sus procesos.
  - La capacidad del conjunto de sus procesos.**
  - El promedio de proyectos exitosos a través de la aplicación del conjunto de sus procesos.
  - Ninguna de las anteriores.
- 
- 8) Respecto al mantenimiento de software:
- Es responsabilidad del Comité de Control de Cambios (CCC) controlar los cambios que se solicitan y aplican a un software que está en Producción.
  - (a) y el CCC debiera integrar la visión de Cliente, Usuarios y Desarrolladores para asegurar que el software evoluciona atendiendo a las necesidades de Cliente y Usuarios y los cambios propuestos son factibles y consistentes.
  - (b) y entre las herramientas que asisten a los desarrolladores en el mantenimiento de software se pueden mencionar las que brindan referencias cruzadas y ayudan a realizar el análisis de impacto.
  - (c) y dejar registros del esfuerzo de mantenimiento correctivo por componente permite evaluar la conveniencia de re-escribir o re-diseñar un componente para mejorar su calidad y reducir los costos de mantenimiento; estas re-escrituras y/o re-diseños constituyen ejemplos de mantenimiento preventivo.**
- 
- 9) Respecto al mantenimiento:
- el mantenimiento correctivo tiene por objetivo corregir defectos del software antes de que estos se manifiesten como una falla. Sin embargo, toda intervención efectuada sobre cierto software con el objetivo de eliminar un defecto, tiene cierta probabilidad de introducir nuevos defectos.
  - Para detectar si como resultado de una intervención se introdujeron nuevos defectos, se suelen llevar adelante pruebas de regresión.
  - (b) y los defectos detectados a partir de las fallas ocurridas durante la explotación del software constituyen una fuente de información relevante respecto a la calidad del software como un todo
  - (c) y de la calidad de cada uno de sus componentes en particular, así como también de la calidad del trabajo de mantenimiento realizado.**
- 
- 10) La Capacidad de un Proceso es:
- Una medida de la madurez de la organización
  - La probabilidad de que los resultados obtenidos luego de su ejecución coincidan con los estimados**
  - Una medida de los resultados reales obtenidos luego de su ejecución
  - La probabilidad de que la madurez del proceso alcance el nivel de optimización
- 
- 11) Los procesos y productos se pueden evaluar con:
- Estudio de casos
  - Experimentos formales
  - Estudios retrospectivos
  - Todas las anteriores**
- 
- 12) Las metodologías ágiles privilegian:
- La respuesta ante el cambio sobre el seguimiento del plan**
  - La planificación sobre la interacción y los individuos
  - El cronograma sobre la verificación de la arquitectura
  - La documentación sobre un software funcionando

- 
- 13) Un proceso de evaluación debe incluir:
- Identificar los requerimientos de calidad
  - (a) y seleccionar cómo serán medidos
  - (b) y definir excepciones a la regla
  - (b) y medir, puntuar y dictaminar**
- 
- 14) La Configuración del Software
- comprende ítems generados durante del proceso de ingeniería de software.
  - (a) y algunos ejemplos pueden ser: especificación de requerimientos, conjuntos de casos de prueba, reportes de defectos, manuales de usuarios.
  - (b) y además esto quiere decir que desde que se crea un elemento de configuración es necesario controlar los cambios mediante procedimientos formales.
  - (b) y un elemento de configuración se convierte en línea base cuando se ha revisado formalmente.**
- 
- 15) El modelo CMMI es
- Un modelo de proceso de desarrollo de software
  - Un modelo de calidad de productos y procesos de software
  - Un modelo de mejora de procesos que recomienda un conjunto de buenas prácticas
  - (c) y que tiene una representación por niveles y otra continua**
- 
- 16) Un antipatrón visto en el curso es:
- Merge-Mania-paranoia: Fusionar continuamente por temor a olvidar una fusión
  - Big Bang Mania: Intentar fusionar todo el tiempo todas las ramas
  - Nunca comenzar a integrar: Aplazar la fusión hasta el final y nunca integrar porque siempre surge un problema
  - Wrong-Way Merge: Fusionar una versión de componentes con una versión obsoleta**
- 
- 17) La verificación:
- Consiste en demostrar la presencia de faltas o, en caso de no detectarse, su ausencia.
  - Es un proceso costoso, pero en la práctica puede seguirse una metodología para probar todas las secuencias de ejecución.
  - Tiene como objetivo descubrir defectos**
  - (c) y corregirlos
- 
- 18) Sobre categorización y registro de defectos
- Puede servir como guía para orientar la verificación en el sentido que si se conoce el tipo de defectos que se inyecta normalmente se los puede buscar de forma específica.**
  - La aplicación de la categorización y registro de defectos solamente es adecuada si se usa un proceso “pesado” de desarrollo, no siendo adecuada en metodologías ágiles.
  - La categorización de los defectos es en tipos de defectos (por ejemplo los vistos en el curso) no siendo necesaria ninguna otra información (como por ejemplo en qué fase se inyectó el defecto).
  - No es conveniente registrar los defectos porque puede provocar importantes demoras en el cronograma de un proyecto.
- 
- 19) El diseño de Casos de Prueba a nivel de sistema a partir de los requisitos del mismo, sin tener el sistema implementado:
- Se puede utilizar para verificar los requisitos
  - (a) y se puede utilizar para estimar el esfuerzo que llevará la ejecución de pruebas de sistema**
  - (b) e implica que se está siguiendo el modelo de desarrollo en V.
  - Ninguna de las anteriores.
- 
- 20) Sobre las técnicas de verificación
- Las inspecciones son una técnica dinámica con alta efectividad en la detección de defectos.
  - El análisis formal, si bien es costoso, lleva a construir sistemas sin defectos.
  - Las técnicas dinámicas son más efectivas en la detección de defectos que las técnicas estáticas
  - Ninguna de las anteriores**
- 
- 21) Sobre las pruebas de integración
- Se realizan luego de implementadas y verificadas todas las unidades (o módulos) que componen el sistema a desarrollar
  - Existen diferentes estrategias de integración como Top-down o Bottom-up**
  - Los drivers son herramientas que automatizan la generación de pruebas de integración
  - En las pruebas de integración nunca se pueden usar pruebas de regresión debido a los cambios que se realizan en el software entre distintas integraciones.

- 22) Sobre el proceso de pruebas unitarias suponiendo que se realizarán pruebas de caja negra y blanca, revisiones personales e inspecciones.
- En el proceso de pruebas unitarias es conveniente primero ejecutar los casos de prueba generados utilizando una técnica de caja blanca y luego de corregidos los defectos que se encuentren complementar con casos de caja negra.
  - Es conveniente primero ejecutar las pruebas unitarias y luego realizar la revisión personal de código.
  - Las inspecciones de las unidades solamente se deben realizar luego de terminadas las pruebas unitarias.
  - En el proceso de pruebas unitarias es conveniente primero ejecutar los casos de prueba generados utilizando una técnica de caja negra y luego de corregidos los defectos que se encuentren complementar con casos de caja blanca.**

- 23) Sobre las pruebas de desempeño
- Es conveniente planificarlas luego de que el sistema cumple con los requerimientos funcionales
  - Incluyen las pruebas estrés y de volumen entre otras**
  - Se deben realizar manualmente ya que no existen actualmente herramientas que den soporte a las mismas
  - Ninguna de las anteriores.

- 24) Sobre quién realiza la verificación
- Las pruebas unitarias son normalmente realizadas por un equipo especializado e independiente del equipo de desarrollo
  - Las pruebas de integración son normalmente realizadas por un equipo especializado e independiente del equipo de desarrollo
  - Para realizar las pruebas unitarias es necesario tener un conocimiento global del sistema y por este motivo las realiza el equipo de desarrollo
  - Las pruebas unitarias son realizadas generalmente por la misma persona que desarrolló la unidad**

- 25) Dado el siguiente método que calcula la Media de los elementos de la lista pasada por parámetro se puede decir que:

```
private static double Media(Lista lista){ //Asumir que es una Lista de elementos double
if ( lista.esVacia() ) return 0; //devuelve true si la lista es vacía, false en caso contrario
int largo = lista.largo();
double total = 0;
while ( lista.masElementos() ) { //devuelve true si quedan elementos en la lista, false caso contrario
    total = lista.primerElemento() + total; //Toma el primer elemento (double) y lo elimina de la lista
}
return total/largo;
}
```

- El dato de prueba lista vacía cumple el criterio de Cubrimiento de sentencias
- El dato de prueba lista vacía cumple el criterio de Cubrimiento de decisión
- El dato de prueba lista con tres elementos cumple el criterio de Cubrimiento de sentencias
- El conjunto de datos de prueba con la lista vacía y con una lista con 1 elemento cumple con el criterio de Cubrimiento de sentencias**