

- 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) Un Comité de Control de Cambios es
 - a. Un grupo conformado por usuarios clave del cliente, cuya función es controlar la correcta implementación de las solicitudes de cambio realizadas sobre los aplicativos liberados.
 - b. Un grupo conformado por usuarios clave del cliente, cuya función es analizar el impacto de realizar un cambio sobre un sistema ya liberado.
 - c. Un grupo compuesto íntegramente por miembros del equipo de desarrollo cuya función es evaluar si corresponde aprobar o rechazar las solicitudes de cambio que provienen del cliente.
 - d. **Un grupo integrado por representantes del cliente, los usuarios y los desarrolladores, cuya función es evaluar y autorizar o rechazar las solicitudes de cambios al sistema, ya sean mejoras o correcciones de defectos.**

- 2) Los sistemas legados:
 - a. Son sistemas desarrollados hace muchos años, para otra realidad, a menudo construidos utilizando tecnologías que ya son obsoletas.
 - b. **(a) y frecuentemente tienen costos de mantenimiento elevados debido a documentación inadecuada, deterioro por años de mantenimiento en los que se priorizaron los resultados a corto plazo por sobre los costos a largo plazo y escasez de personal que en cada caso conozca las tecnologías utilizadas en la implementación y el sistema en particular.**
 - c. (b) y les son aplicables las técnicas de rejuvenecimiento, reestructuración de código, re-documentación, e ingeniería reversa, no así la de re-ingeniería que consiste en la reestructuración del código seguida por una re-documentación del código reestructurado.
 - d. Normalmente son fáciles de sustituir por sistemas más modernos mediante una estrategia de implantación big-bang la que -a diferencia de las estrategias paulatinas- permite obtener una rápida reducción en los costos de mantenimiento.

- 3) En el mantenimiento de software:
 - I. Los encargados de llevar a cabo el mantenimiento suelen tener una comprensión limitada del sistema que están manteniendo, por lo que gran parte del esfuerzo se dedica a entender el sistema bajo mantenimiento.
 - II. Realizar el análisis del impacto de un cambio representa normalmente una porción muy pequeña del esfuerzo total dedicado a realizar el cambio.
 - III. Es frecuente que la gerencia asigne más importancia al mantenimiento que al desarrollo, lo que resulta en un importante factor de motivación para el personal encargado del mantenimiento.
 - IV. La realización de las pruebas requiere contar con un ambiente adecuado para llevarlas a cabo, lo que suele plantear dificultades, sobre todo si se requiere contar con datos similares a los del ambiente de producción.
 - V. Por lo general existe un conflicto entre la necesidad de contar rápidamente con una solución y que la solución preserve atributos de calidad relevantes del producto.
 - VI. La secuencia de reparaciones y mejoras centradas en soluciones de corto plazo permiten “rejuvenecer” al software, mejorando sus atributos de calidad y en especial la facilidad de su mantenimiento.

De las afirmaciones anteriores son correctas las identificadas por los números:

 - a. I, II, III, IV, V, VI
 - b. I, III, IV, V
 - c. II, III, V
 - d. **I, IV, V**

- 4) Al liberar un sistema, en general se realiza un entrenamiento del mismo:
 - a. Normalmente se debe prever el entrenamiento para los usuarios finales, no así para los operadores y administradores que suelen estar en condiciones de aprender el uso del sistema por cuenta propia.
 - b. **Resulta conveniente evaluar la efectividad del entrenamiento en base al grado de cumplimiento de los objetivos establecidos para el mismo y del grado de uso del sistema y la eficiencia que logran los usuarios al utilizarlo una vez que ya fue puesto en explotación.**
 - c. El entrenamiento no debe tomar en cuenta características y preferencias personales ni estilos de trabajo.
 - d. El entrenamiento conviene que se imparta en etapas tempranas del proyecto a todos los usuarios.

- 5) Dado un proyecto para sustituir el sistema que da soporte al funcionamiento de una organización por un sistema nuevo y que incluye análisis, diseño, construcción, prueba y puesta en producción (liberación) del software:
- La planificación de la liberación debiera comenzar una vez que el producto fue integrado de forma exitosa y antes que comience la prueba del sistema.
 - La estrategia de liberación más conveniente para manejar riesgos asociados a la introducción de nuevos procedimientos es la big-bang.
 - La sustitución probablemente requerirá migrar información del sistema viejo al nuevo y esta conversión puede requerir un plazo y esfuerzo muy considerable.
 - De forma previa a la puesta en producción se podría realizar un procesamiento en paralelo durante una o dos semanas, lo que constituye un mecanismo de verificación del software sumamente efectivo.

- 6) El mantenimiento “Adaptativo” implica
- Dar respuesta a los problemas que surgen en el uso diario.
 - Dar respuesta a los cambios que ocurren en el ambiente de aplicación del software y en las reglas del negocio.
 - Mejorar algún aspecto o funcionalidad ya presente en el software
 - Introducir cambios para prever (o anticipar) fallas en el sistema.

- 7) Una organización dedicada al mantenimiento de software registra información asociada a la actividad de mantenimiento de los distintos productos. Para uno de los productos se ha dedicado en el último año un esfuerzo total de 21528 horas-persona y el 70% de ese esfuerzo se concentra en 5 componentes. En el cuadro siguiente se presenta el esfuerzo en horas-persona dedicado a cada uno de esos componentes durante ese período, discriminado por tipo de mantenimiento.

| Tipo de Mantenimiento | Correctivo | Perfectivo | Adaptativo | Preventivo | Total |
|-----------------------|------------|------------|------------|------------|-------|
| Componente | | | | | |
| A | 100 | 3000 | 2000 | 0 | 5100 |
| B | 1100 | 2300 | 350 | 0 | 3750 |
| C | 50 | 0 | 3020 | 0 | 3070 |
| D | 2100 | 0 | 0 | 0 | 2100 |
| E | 230 | 300 | 120 | 400 | 1050 |
| Total | 3580 | 5600 | 5490 | 400 | 15070 |

A partir de los datos del cuadro, sería conveniente considerar y evaluar con más detenimiento:

- Reescribir el componente A, que es al que se ha destinado mayor esfuerzo de mantenimiento.
- Reescribir el componente E, que de los 5 es al que se ha destinado mayor esfuerzo de mantenimiento preventivo.
- Reescribir los componentes B y D, que de los 5 son en los que se ha concentrado la mayor parte del esfuerzo de mantenimiento correctivo.
- Destinar esfuerzo al mantenimiento perfectivo, adaptativo y preventivo del componente D, que solo ha recibido mantenimiento correctivo.

- 8) Son objetivos de la verificación y validación de software:
- Descubrir faltas, tanto sea provocando defectos o revisando el producto.
 - Evaluar la calidad del código fuente y/o documentos generados durante un proyecto de software.
 - (b) y descubrir defectos, por ejemplo, mediante la revisión de documentos o código fuente.
 - (c) Luego de cumplido estos objetivos, el proceso de verificación y validación culmina con la corrección de los defectos detectados.

- 9) Respecto a la verificación y validación
- El proceso de requisitos visto en el curso no tiene actividades de verificación o validación ya que el mismo se enfoca en obtener y especificar los requisitos.
 - Registrar y categorizar los defectos detectados sirve para mejorar el proceso de desarrollo de software; por ejemplo, definiendo estrategias de prevención de defectos.
 - (b) Sin embargo, no se deben registrar los defectos encontrados en las pruebas unitarias ya que serían demasiados a registrar y esto aumentaría el costo (tiempo en horas del registro de defectos) de forma importante.
 - Registrar y categorizar los defectos detectados durante el desarrollo de software es tan costoso que el costo-beneficio se vuelve casi nulo

-
- 10) Respecto a la verificación y validación
- Uno de los objetivos consiste en probar la ausencia de defectos, esto es, que el sistema no falla y está libre de errores.
 - Las formas de detección de defectos se pueden clasificar en dos: una es hacer que el software falle y la otra consiste en demostrar formalmente que un programa es correcto.
 - La verificación se realiza en todas las fases de desarrollo (requerimientos, análisis, diseño, implementación), y consiste en evaluar si se está construyendo el producto correctamente.
 - Si como resultado de la verificación no se detectan defectos, entonces se considera que el sistema fue correctamente verificado y validado, y que en ese caso no es necesario realizar la validación por parte del cliente.
-
- 11) En la generación de casos de prueba a partir de casos de uso
- Se deben tener en cuenta los flujos alternativos del CU
 - (a) y muchas veces es necesario considerar los flujos alternativos más de una vez (en distintos escenarios) y en distintas combinaciones con otros flujos alternativos
 - (b) y los flujos alternativos globales, que pueden activarse en cualquier punto del CU (p.e. "G1. Cancelar la operación") deben probarse una única vez.
 - (c) y es necesario probar al menos una vez cada escenario generado del CU.
-
- 12) Respecto al proceso de verificación de software
- Las pruebas unitarias de software son realizadas normalmente por el equipo de desarrollo y casi siempre por el propio desarrollador de la unidad.
 - (a) y las técnicas estáticas tienen una menor efectividad (cantidad de defectos detectados dividido la cantidad de defectos del producto) que las técnicas dinámicas.
 - Es normal que todas las pruebas de software las realice un equipo especializado, mientras que las técnicas estáticas (ej: revisión, inspecciones) las realiza el equipo de desarrollo.
 - (c) y el proceso visto en el curso para la verificación contiene fases para la verificación unitaria, las pruebas de integración y distintas pruebas a nivel de sistema (funcional, desempeño, etc).
-
- 13) En cuanto a la verificación de software, indique cuál de las siguientes afirmaciones **NO** es correcta.
- Las pruebas exhaustivas son las únicas técnicas dinámicas que podrían asegurar la correctitud de un producto de software.
 - Considerando las pruebas de caja blanca, las técnicas de pruebas basadas en el flujo de datos son más usadas que las técnicas de pruebas basadas en el flujo de control.
 - Existe una técnica de pruebas de integración llamada sándwich.
 - La prueba de programas puede usarse para mostrar la presencia de defectos, pero no para demostrar su ausencia.
-
- 14) Respecto a la verificación y validación
- El proceso de verificación y validación es poco costoso respecto a los otros sub-procesos del desarrollo de software.
 - La planificación de la V&V conviene comenzar a realizarla cuando el producto está en una fase de codificación ya que antes es totalmente innecesario (se está muy lejos de la fase de verificación).
 - Un buen criterio de terminación para las pruebas de sistema es culminarlas cuando el tiempo adjudicado en el plan de V&V para estas pruebas se ha terminado.
 - Una forma de estimar la cantidad de defectos remanentes en el software es utilizar a dos grupos independientes para que prueben el software. Conociendo la cantidad de defectos encontrados por ambos grupos y la cantidad de defectos encontrados en común se puede estimar la cantidad de defectos remanentes.
-
- 15) Se encuentra la siguiente descripción de un sistema en la revista "Pequeños Sistemitas Funcionando" de la CPAP Informática del 12 de octubre de algún año: "El sistema deberá soportar 130 usuarios conectados a la vez. El mismo deberá poder ejecutarse tanto en Linux como en Windows con Oracle o Informix como manejadores de bases de datos". Probar el sistema para Windows con Oracle, Windows con Informix, Linux con Oracle y Linux con Informix es una prueba (principalmente) de:
- estrés (esfuerzo)
 - configuración
 - volumen
 - de aceptación
-
- 16) La nueva norma ISO/IEC 25000, reúne:
- Varias normas de calidad de productos de software.
 - La norma de calidad de software ISO/IEC 9126 y la norma ISO/IEC 15504 de evaluación de procesos.
 - La norma de calidad de software ISO/IEC 9126 y la norma ISO/IEC 14598 de evaluación del software.
 - (b) y la familia de normas ISO 9000.

-
- 17) Sobre los procesos de software:
- Son un conjunto de actividades, métodos, prácticas y transformaciones que siempre son evaluados buscando la mejora continua.
 - El desempeño en la ejecución de un proceso es una medida de los resultados reales conseguidos como efecto de su realización.
 - (b) y es importante que las organizaciones logren anticipar el desempeño de sus procesos.**
 - (c) y la capacidad de un proceso y la madurez de la organización son formas de medir este desempeño.
-
- 18) ¿Cuál de estas afirmaciones sobre las pruebas de integración es **INCORRECTA**?
- La técnica Big Bang no necesita de verificación unitaria ya que se integran todos los módulos a la vez. Por esto mismo no se necesitan ni stubs ni drivers.**
 - La técnica Bottom-Up utiliza la relación “un módulo usa otro módulo” y comienza por los módulos “de abajo” (aquellos que no usan a otros módulos) y va “subiendo” en la relación mencionada, integrando los módulos.
 - La técnica por disponibilidad va integrando los módulos que van siendo liberados por el equipo de desarrollo.
 - Ninguna es incorrecta.
-
- 19) La diferencia entre los estudios retrospectivos y los experimentos formales radica en:
- Los primeros estudian ciertas variables en proyectos ya finalizados.
 - (a) A su vez, los experimentos formales planifican un proyecto para validar o refutar una hipótesis,
 - (b) definiendo y controlando ciertas variables consideradas relevantes.**
 - Ninguna de las anteriores.
-
- 20) Las áreas de proceso del modelo CMMI se agrupan:
- Por nivel de la representación continua del modelo.
 - Por categorías de procesos que son: ingeniería, gestión de proyectos,
 - (b) gestión de procesos y soporte.**
 - Por afinidad de los aspectos considerados en cada área.
-
- 21) Con respecto a la Gestión de la Configuración (SCM):
- Identifica, organiza y controla las modificaciones de los artefactos de software (teniendo en cuenta que esto comprende documentación, código fuente, código ejecutable y todo artefacto intermedio) a lo largo de su ciclo de vida.
 - (a) y algunas actividades de SCM son Identificación de elementos y Control de versiones.
 - (b) y en la identificación de elementos cada elemento se identifica de forma única.**
 - (c) y para el Control de versiones no es posible versionar asociando un número a cada versión.
-
- 22) Las metodologías ágiles:
- Promueven la construcción de software en tiempos muy reducidos y sin documentación.
 - Privilegian los individuos y sus interacciones sobre procesos y herramientas.
 - (b) y la respuesta al cambio sobre el seguimiento del plan y un software funcionando sobre la documentación.
 - (c) y la colaboración con el cliente sobre la negociación de contratos.**
-
- 23) La característica del software “*Confiabilidad*” se define, según el modelo ISO/IEC 9126 como:
- Un conjunto de atributos relacionados con la capacidad del software para mantener su nivel de desempeño en determinadas condiciones por un período de tiempo.**
 - Un conjunto de atributos concerniente a la relación entre el nivel de desempeño del software y la cantidad de recursos consumidos.
 - Un conjunto de atributos relacionados con el esfuerzo de uso para un grupo de usuarios.
 - Un conjunto de atributos relacionados con la existencia de un conjunto de funciones y sus propiedades específicas.
-
- 24) El análisis post-mortem de un proyecto es útil para:
- Identificar los culpables de los atrasos y problemas surgidos durante el proyecto.
 - Identificar los aciertos y dificultados surgidos durante el proyecto.**
 - Estudiar principalmente el desempeño de los recursos humanos involucrados en el proyecto.
 - Todas las anteriores.
-
- 25) Sobre la gestión de la configuración, indique cuál de las siguientes afirmaciones es **INCORRECTA**:
- El Comité de Control de Cambios es quién evalúa y aprueba los cambios en los distintos elementos de configuración.
 - La Gestión de la Configuración de Software (SCM) se encarga de la gestión de la configuración del software, pero también de la gestión de la configuración de la documentación asociada (por ejemplo: manual de usuario).
 - El responsable de SCM es quién resuelve las solicitudes de cambio.**
 - En proyectos grandes es importante contar con herramientas que apoyen la gestión de la configuración.
-