

- Cada pregunta **múltiple opción** contestada correctamente tiene un valor de 3,34 puntos. Cada pregunta incorrecta de la múltiple opción resta 1,11 puntos.
- **El puntaje total del examen es 100 puntos y se aprueba con 60 o más puntos.**

Múltiple Opción

1. Respecto a la ingeniería de software:
 - I. Es una disciplina que ha presentado muchos cambios durante los últimos veinte años, impulsados entre otros factores por la sostenida baja relativa del costo del hardware respecto al del software, el desarrollo de redes (en particular Internet) y la creciente presión por reducir el plazo de salida al mercado (time to market).
 - II. La Gestión de Recursos Humanos no es muy importante en la ingeniería de software dado que la construcción de software es una rama industrial que hace un uso muy intensivo de tecnología y en términos relativos la mano de obra tiene poca incidencia en el producto final.
 - III. Está relacionada con la ingeniería de sistemas, dado que al construir software en general estamos construyendo a la vez un sistema compuesto al menos por software, hardware sobre el que ejecuta el software y personas que lo utilizan y es ese sistema complejo el que debe funcionar y el relevante para evaluar la satisfacción del cliente/usuario.
 - IV. Está relacionada con la programación, pero construir un producto conformado por un sistema de programas que deben interactuar entre sí de forma consistente abarca diversidad de aspectos que están más allá de la programación. Desde el punto de vista del esfuerzo necesario en relación al tamaño, resulta bien distinto construir:
 - Un programa
 - Un producto de software
 - Un sistema de programas
 - Un producto sistema de programas.Para un mismo tamaño, construir un producto de software requiere más esfuerzo que construir un programa. Construir un sistema de programas requiere más esfuerzo que la suma de los esfuerzos necesarios para construir cada uno de los programas que lo componen y construir un producto constituido por un sistema de programas requiere más esfuerzo que el necesario para construir el sistema de programas.
 - V. Persigue la mejora de la productividad y calidad en la producción de software fundamentalmente a través de la introducción de herramientas que automaticen los procesos de recopilación de requerimientos, diseño, codificación y verificación.
 - VI. El éxito del software en el sentido de que está omnipresente y la creciente dependencia del funcionamiento de la sociedad respecto al funcionamiento del software trae aparejado una creciente preocupación social por asegurar la calidad del software mediante normas, reglamentaciones y regulaciones.

Son correctas las respuestas:

- a) I, III, IV, VI
 - b) II, V
 - c) I,II,III, IV,V,VI
 - d) II, III, IV,VI
-
2. Sobre patrones de software:
 - a) Tienen como objetivo compartir la solución de un problema complejo.
 - b) Documentan soluciones probadas de problemas conocidos.
 - c) "Idioms", "Estilos modulares" y "Observer" son ejemplos de patrones de diseño.
 - d) (b) y (c) son correctas.
-
3. En el enfoque de valor ganado, el Índice de Desempeño del Cronograma ($SPI = VG/VP$):
 - a) Indica cuán adelantados o atrasados estamos respecto a la planificación.
 - b) Indica cuán eficientemente estamos usando los recursos.
 - c) Está indicado por la pendiente de la línea de VG.
 - d) Indica la velocidad a la que vamos ganando valor.
-
4. Luego de culminado el diseño arquitectónico de un sistema y durante el diseño detallado de algunas componentes surgen cambios en los requisitos que influyen en la arquitectura definida. ¿Cuáles son los pasos a seguir?
 - a) Es recomendable que el diseño se base en la arquitectura definida para el sistema. Sin embargo, no importa si luego el diseño se desvía de la misma por algún motivo válido. Entonces, en este caso no importaría analizar la arquitectura para conocer en qué influyen exactamente los nuevos requisitos y cómo contemplarlos.
 - b) Se deben analizar los impactos que puedan tener los cambios solicitados.
 - c) (b). La trazabilidad ayuda a conocer qué componentes de la Arquitectura se pueden ver afectados por los cambios..
 - d) (c). No es necesario analizar impactos en los casos de prueba aunque estos ya hayan sido definidos, ya que la verificación del sistema aún no se ejecutó.

-
5. 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.
 - (b) y los drivers son herramientas que automatizan la generación de pruebas de integración.
 - (b) y en las pruebas de integración se pueden usar pruebas de regresión.
-
6. Las normas de calidad de un producto de software, como la ISO/IEC 9126 o la más reciente 25000 son útiles para:
- Describir características y atributos de un producto de software.
 - Definir los objetivos de calidad de un producto de software previo a su construcción.
 - Evaluar un producto de software para su aceptación.
 - Todas las anteriores.
-
7. Al relevar requerimientos:
- se deben buscar antecedentes,
 - (a) y por antecedentes se entienden documentos existentes
 - (b) y sistemas análogos existentes en el mercado
 - (c) y por ejemplo leyes que determinen el funcionamiento del sistema
-
8. La técnica de revisión de requisitos puede ser usada para validar los mismos. En ese caso la revisión incluye:
- Verificar la trazabilidad de los requisitos con los casos de uso.
 - Revisar si las funcionalidades están completas (que estén todas, que no haya omisiones, que estén analizadas las respuestas a todas las posibles entradas, etc.), y si las restricciones son realistas.
 - Pensar en el diseño necesario para concretar los requisitos.
 - Todas las anteriores.
-
9. Dados los siguientes proyectos y modelos de proceso:
- Proyecto:
- A - de alto riesgo, muy complejo, alcance no del todo definido, subconjuntos de funcionalidad definidos que resultan útiles para los usuarios, larga duración (dos años o más).
- B - Riesgo medio, alcance definido, subconjunto de funcionalidad definido y útil para los usuarios, duración media (seis meses).
- C - Riesgo bajo, simple, alcance definido, corta duración (seis semanas).
- Proceso:
- 1 - Cascada
- 2 - En Fases con Evaluaciones Parciales Internas
- 3 - En Fases con Liberaciones Parciales en Producción
- 4 - De Prototipación
- 5 - Especificación Operacional
- 6 - En Espiral
- Marque la asignación que le parezca más adecuada. Si un mismo proyecto aparece asignado a más de un proceso, debe entenderse que cualquiera de los dos resultaría adecuado.
- Para los procesos 2 y 3 se puede especificar además entre paréntesis el modelo de proceso a utilizar en cada fase. Por ejemplo 2(1) significa Proceso en Fases con Evaluaciones Parciales Internas y en cada fase se aplica Cascada. 3(2(1)) significa que en este caso en cada fase de 3 se aplica En Fases con Evaluaciones Parciales Internas y en cada fase de 2 se aplica Cascada.
- A3(2(1)), A4, A5, A6, B3(2(1)), B3(1), C1, C2(1), C4
 - A1, A2(1), A3(2(1)), A4, B3(1), B4, C1
 - A3(2(1)), A6, B3(2(1)), B3(1), C1, C2(1), C6
 - A3(2(1)), A6, B3(2(1)), B3(1), C1, C2(1), C4
-
10. En verificación de software:
- Una falla no siempre genera una falta.
 - Una falta puede generar una falla.
 - Los errores humanos siempre generan fallas.
 - Más de una opción es correcta.
-

11. Respecto a la liberación de software:
- I. Es una etapa que si no se ha planificado de forma oportuna y/o destinado recursos adecuados puede ser fuente de problemas para los proyectos y generar insatisfacción en usuarios o cliente. La planificación de esta etapa debiera comenzar en la etapa inicial del proyecto en la que se define el alcance del mismo.
 - II. Se debe prever eventual necesidad de entrenamiento, documentación para los distintos tipos de usuarios, eventual necesidad de conversión de datos de un sistema anterior al nuevo, instalación en ambiente objetivo y mecanismos para la solución de problemas.
 - III. La planificación de la liberación debiera comenzar una vez que el proyecto dispone de una versión beta que pueda ser evaluada por el cliente.
 - IV. Al liberar un producto de software los problemas que pueden aparecer están relacionados fundamentalmente con defectos del software y rara vez aparecen problemas con otro origen.
 - V. La calidad tanto de la documentación para el usuario como la del entrenamiento pueden tener impacto sobre la cantidad y entidad de los problemas que aparecen en la liberación.
 - VI. La mejor estrategia de liberación es la paulatina, ya que permite reducir riesgos y no presenta ninguna desventaja respecto a la liberación big-bang.

Son correctas las respuestas:

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

-
12. Una poscondición en un caso de uso:
- a) establece estados en los que queda el sistema al completarse el caso de uso.
 - b) establece qué cosas deben ser siempre verdaderas antes de comenzar un caso de uso.
 - c) es una condición que debe verificarse dentro del caso de uso, pero que no se asume verdadera desde el comienzo.
 - d) establece los requisitos no funcionales que deben verificarse al realizar el caso de uso.

-
13. Considere los siguientes tipos de Mantenimiento:
- I. Correctivo, corresponde en dar respuesta a los problemas que surgen en el uso diario.
 - II. Adaptativo, corresponde a dar respuesta a cambios en el ambiente.
 - III. Perfectivo, corresponde a mejorar algún aspecto ya presente.
 - IV. Preventivo, corresponde a introducir cambios para prever posibles fallas.

Las siguientes opciones son correctas:

- a) Sólo I) y II) son correctas
- b) Sólo I), II) y III) son correctas
- c) Sólo II), III) y IV) son correctas
- d) Todas las opciones son correctas

-
14. En relación a la gestión de la configuración, el alcance junto con el formalismo permiten que el orden que introduce la disciplina de la gestión de la configuración se pueda llevar a la práctica. Indique la opción correcta de las siguientes afirmaciones relacionadas a ese tema.
- a) Esto se debe a que con un único alcance sería imposible trabajar.
 - b) El concepto de alcance corresponde a un conjunto arbitrario de ítems de configuración para el cual se establece un mismo formalismo.
 - c) El alcance es el negociado con el cliente luego de relevados los requisitos del producto.
 - d) Dos de las anteriores son verdaderas.

-
15. Respecto a modelos de proceso:
- a) El esquema ETVX (Entry, Task, Verification, eXit) se usa como plantilla para especificar modelos de proceso. Para cada tarea se especifican los elementos de entrada (Entry), los elementos de salida (eXit) y el procedimiento de verificación (Verification) de que se obtuvo el resultado esperado. Esto último es importante para aclarar el significado de la tarea mediante la especificación de atributos del resultado y para evitar que haya una diferencia significativa entre el "avance percibido" y el "avance real" de un proyecto.
 - b) La principal ventaja del modelo de proceso en fases con evaluaciones parciales internas es que permite reducir el plazo para salir al mercado.
 - c) La principal ventaja del modelo de proceso en cascada es que permite una retroalimentación temprana por parte de los usuarios a partir de su evaluación de los atributos de calidad externos del producto.
 - d) La construcción de prototipos persigue el objetivo de evaluar de forma temprana atributos de calidad de la interfaz de usuario. Normalmente no es posible evaluar otros atributos de calidad tales como tiempo de respuesta o consumo de recursos a partir de un prototipo.

-
16. Ud. es el gerente un proyecto de desarrollo de software. Durante la fase de validación, el cliente requiere de un cambio en uno de los requisitos más importantes, que puede impactar negativamente en el alcance, plazos y costos del proyecto. El cliente requiere además no retrasar la fecha de finalización del proyecto. ¿Cuál le parece a Ud. la mejor estrategia para gestionar este cambio?
- Si es lo que el cliente solicita, entonces el cambio se debe aceptar a cualquier costo.
 - Se analiza el impacto que tiene el cambio sobre el alcance, plazos y costos del proyecto, y se lleva a cabo el mismo.
 - Se realiza una estimación del impacto del cambio solo en términos de costos (ya que los tiempos no pueden variar), y se valida con el cliente. Si está de acuerdo se acepta y se lleva a cabo.
 - Se analiza el impacto del cambio. Ya que el plazo debe mantenerse fijo, entonces de acuerdo a la triple restricción el costo y alcance del proyecto podrán ser impactados. Luego se negocia con el cliente si se cambia el alcance y/o se incrementa el costo.
-
17. Entre las cualidades más relevantes para un software que ofrezca servicios de realidad aumentada para personas no videntes a partir de la ubicación del usuario en un teléfono celular inteligente, están:
- Seguridad de la integridad física (safety), adecuación al uso, eficiencia en el uso de recursos, facilidad de uso.
 - Seguridad de la información, adecuación al uso, Facilidad de mantenimiento, Calidad de la documentación técnica.
 - Modularidad, Facilidad para ser probado, Facilidad de mantenimiento, Legibilidad del código.
 - Disponibilidad, Adecuación al uso, Facilidad de mantenimiento, Facilidad para ser probado.
-
18. En cuanto a los factores que afectan la comunicación en un grupo:
- El tamaño del grupo tiene un efecto proporcional en la comunicación. Las líneas de comunicación se multiplican de forma directamente proporcional a la cantidad de integrantes.
 - Es importante minimizar las áreas físicas comunes del personal, dado que el 50% del tiempo se pierde en interacciones.
 - No solo afectan la comunicación el tipo de personalidad de cada integrante del grupo por separado, sino también la combinación de estos.
 - La comunicación es más fluida y da mejores resultados en un equipo democrático o "egoless".
-
19. Según el proceso de Requisitos visto en el curso:
- Luego de iniciada la actividad de obtención de requisitos y paralelamente con ella es necesario llevar adelante la actividad de análisis de los mismos.
 - El análisis de requisitos es responsabilidad de clientes, usuarios y stakeholders.
 - El análisis de requisitos consiste en evaluar la factibilidad técnica y evaluar los riesgos del proyecto entre otros.
 - Todas las opciones son correctas.
-
20. Con respecto al mantenimiento de software:
- El costo total de mantenimiento a lo largo de todo el ciclo de vida de un producto normalmente resulta despreciable respecto al costo de desarrollo.
 - Normalmente el costo de mantenimiento de los productos de software se mantiene estable, por más que no se haga nada especial para reducirlo, como resultado de que a medida que pasa el tiempo y se corrigen los defectos que van apareciendo en producción, la densidad de defectos del código tiende a decrecer.
 - Frecuentemente aparece un conflicto entre las necesidades de corto plazo (solucionar un problema) y las necesidades de largo plazo (mantener un producto con determinados atributos de calidad).
 - En general la gerencia asigna mayor importancia al mantenimiento de software que al propio desarrollo, lo que constituye un importante factor de motivación del personal de mantenimiento.
-
21. Respecto de la verificación y la calidad del software:
- Los resultados de la verificación sirven para medir algunos aspectos de la calidad de software
 - Los resultados de las evaluaciones de la calidad del software no toman en cuenta los resultados de las pruebas dinámicas
 - Existen atributos de calidad del software que no es posible evaluar mediante pruebas de software
 - Calidad y verificación son dos disciplinas independientes y solo evalúan diferentes aspectos del software
 - Uno de los objetivos de verificación es evaluar la calidad del software
- I), II) y III) son verdaderas y IV) y V) son falsas.
 - II), III) y V) son verdaderas, I) y IV) son falsas.
 - I), IV) y V) son verdaderas, II) y III) son falsas.
 - I), III) y V) son verdaderas, II), IV) son falsas.
-

-
22. Sobre las técnicas de modelados y especificación de requisitos:
- La técnica de modelado de casos de uso es indicada para cualquier tipo de requisito, ya que es la que explica paso a paso cómo es que el usuario y el sistema interactúan entre sí.
 - La técnica de máquinas de estado no es muy usada, ya que la técnica de modelado de casos de uso es más detallada que ésta.
 - La técnica de tormenta de ideas es una técnica de especificación de requisitos que consta de una fase de generación y otra de reducción.
 - En algunos casos es aconsejable especificar todos los requisitos en lenguaje natural en un documento de requisitos, y, eventualmente, volver a especificar algunos de ellos con otra técnica más gráfica de acuerdo al tipo de requisito que se esté tratando.
-
23. Los estudios retrospectivos son una técnica de evaluación utilizada para:
- Analizar las tendencias y las relaciones entre varios productos y/o proyectos.
 - Validar o refutar hipótesis sobre productos y/o proyectos.
 - Comparar los resultados de varios proyectos con la línea base previamente establecida.
 - Ninguna de las anteriores.
-
24. Sobre diseño de software. El acoplamiento:
- Es un indicador sobre la dependencia entre módulos.
 - (a). Además, el alto acoplamiento facilita la comprensión y mantenimiento de los módulos del sistema.
 - (a) Además, el bajo acoplamiento facilita la especificación de los casos de uso mediante la reducción de flujos alternativos.
 - Es un indicador de qué tan cohesivo es un módulo
-
25. Sobre la gestión de la configuración:
- El formalismo está relacionado al nivel de control y estructurados que son los procedimientos que ordenan y controlan mi configuración.
 - (a) y para cada proyecto hay un único formalismo y está definido por la gestión de proyectos.
 - (a) y es posible tener distintos niveles de formalismos según cada alcance en un mismo proyecto u organización.
 - (c) y además también es posible definir diferentes niveles de formalismo para alcances no disjuntos.
-
26. El diseño de Casos de Prueba a nivel de sistema a partir de los requisitos del mismo, sin tener el sistema implementado:
- Es excelente para planificar la Inspección de software (artefactos a inspeccionar, duración de las inspecciones, etc.)
 - Se puede utilizar para estimar el esfuerzo que llevará la ejecución de las pruebas de sistema.
 - (b) y No es conveniente comenzar a desarrollar la arquitectura de software antes de tener el diseño de casos de prueba de sistema totalmente finalizado.
 - Se debería realizar a partir de los casos de uso cuando el sistema aún no fue implementado y no a partir del documento de especificación de requisitos.
-
27. Dadas las siguientes afirmaciones indique cuál respuesta es correcta.
- Descubrir defectos es uno de los objetivos de la verificación.
 - Evaluar la calidad de los productos es uno de los objetivos de la verificación.
 - La corrección de los defectos detectados es parte de la verificación.
 - Es habitual en la industria de software que las pruebas unitarias las realice un equipo especializado de testing y no el equipo de desarrollo.
 - El conocimiento de las interfaces y la forma en la que interactúan las componentes de un sistema son importantes para realizar las pruebas de integración.
 - La Inspección de software no es muy usada en la industria de software debido a que su costo/beneficio no es razonable (es decir, es muy costosa para los beneficios que se logran)
- Se cumple 1 pero no se cumple 2.
 - Se cumplen 1, 2 y 4 pero no se cumple 5.
 - Se cumplen 1, 2, 5 y 6 pero no se cumple 3.
 - Se cumplen 1, 2 y 5 pero no se cumplen 3 y 6.
-
28. Los requisitos describen:
- las funcionalidades y servicios que proveerá el sistema.
 - la interacción del sistema con su entorno.
 - las restricciones que limitan la elección de la solución.
 - todos los anteriores.
-

29. Para que una empresa de construcción de software se certifique Nivel 3 de CMMI (representación por niveles) debe:
- Cumplir con todos los atributos de las áreas de proceso de Ingeniería.
 - Cumplir con los atributos de las áreas en las que desea desarrollarse hasta ese nivel.
 - Tener sus procesos gestionados, definidos e implementados para toda la organización.
 - Tener algunos procesos a Nivel 1, otros a Nivel 2, pero los más importantes a Nivel 3.
-
30. Sobre la verificación de software
- Las pruebas de integración no tiene sentido realizarlas si se tiene un equipo de desarrollo que realiza excelentes pruebas unitarias y un equipo especializado en pruebas de sistema.
 - Las Inspecciones han mostrado ser una técnica más costosa que la técnica de pruebas de trayectorias linealmente independientes.
 - La ejecución simbólica es utilizada extensamente en la industria de software. En particular en el desarrollo de sistemas de información.
 - En las pruebas de caja blanca se utiliza el código de los programas para generar los casos de prueba.