

- 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 cambios en procesos de desarrollo de software:
  - a) En general los procesos luego de definidos e implementados no se modifican.
  - b) Un cambio en un proceso puede incluir las siguientes tareas: introducir nuevas prácticas, métodos o herramientas; cambiar el orden de las actividades; introducir nuevos roles y responsabilidades.
  - c) b) y luego de introducidos los cambios en los procesos estos deben ser difundidos a los involucrados y en caso de ser necesario realizar el entrenamiento correspondiente.
  - d) c) y cuando existen varios cambios para realizar a un proceso se aconseja hacerlos todos juntos.

---

2. Con respecto al mantenimiento de software:
  - a) El software requiere mantenimiento si presenta defectos. Si se lograra producir software libre de defectos, prácticamente no sería necesario realizar tareas de mantenimiento.
  - b) Normalmente la solución más simple para la implementación de un cambio es a la vez la más rápida de implementar y la que mejor preserva los atributos de calidad relevantes del producto por lo que rapidez y calidad de la solución suelen ir de la mano.
  - c) A medida que se van acumulando intervenciones de mantenimiento correctivo sobre el software, de manera análoga a lo que sucede en desarrollo, la calidad del software tiende a mejorar dado que la cantidad de defectos remanentes normalmente decrece de forma sistemática.
  - d) Entre los problemas técnicos asociados al mantenimiento de software está el disponer de un ambiente de pruebas adecuado, en el que sea posible reproducir problemas encontrados en producción.

---

3. Con respecto a la arquitectura de software:
  - a) Por lo general la información relevante de la arquitectura se puede representar con una única vista.
  - b) En el modelo 4+1, la vista de desarrollo muestra las abstracciones clave del sistema, por ejemplo: la distribución de las clases en los nodos físicos.
  - c) b) y en las arquitectura distribuidas el procesamiento de la información es distribuido entre varias computadoras.
  - d) Ninguna de las anteriores es correcta.

---

4. Con respecto a la arquitectura de software:
  - a) Un idiom, es un patrón de bajo nivel, específico para un lenguaje de programación.
  - b) Un estilo arquitectónico, provee un esquema para refinar los elementos de un sistema de software o las relaciones entre ellos.
  - c) Un patrón de diseño, expresa un esquema de organización estructural a nivel de sistemas de software.
  - d) Todas son correctas.

---

5. Respecto a la liberación de software:
  - a) Los aspectos que corresponde considerar relacionados con la liberación de software son: el entrenamiento a los usuarios, la documentación del software, la solución de problemas asociados a su puesta en funcionamiento, la conversión de un sistema anterior al nuevo (en aquellos casos en que el software sustituye a un sistema previo), y lo atinente a la instalación del software.
  - b) Es una etapa de los proyectos de software a la que en general no vale la pena destinar mucho esfuerzo, ya que difícilmente pueda tener impacto significativo en la evaluación de los resultados por parte de cliente y usuarios.
  - c) La facilidad de instalación del software es un atributo de calidad del mismo que solo es relevante en el momento en que el software se pone en producción por primera vez.
  - d) Los únicos tipos de problemas que pueden aparecer inmediatamente luego de la liberación del software son aquellos que están relacionados con defectos del software, por lo que si se llevó a cabo una adecuada verificación del mismo, normalmente no resulta necesario prever recursos para la solución de problemas asociados a la liberación.

---

6. Son actividades de SCM:
  - a) Gestión de Cambios.
  - b) a) y gestión de versiones.
  - c) Auditar el desarrollo.
  - d) Todas las anteriores.

7. Respecto a modelos de proceso:
- Una ventaja de modelo de proceso en fases con evaluaciones parciales internas es que el cliente puede contar con resultados de utilidad en un plazo más corto, lo que permite reducir el plazo de salida al mercado (time to market).
  - La única ventaja del modelo de proceso en fases con liberaciones parciales es que permite contar con retroalimentación temprana de parte tanto de cliente como de usuarios acerca de los atributos de calidad externos del producto.
  - El modelo de proceso en espiral se puede ver como un proceso en cascada al que se le agrega el análisis de riesgos, aspecto que cada vez resulta más relevante, por lo que este modelo de proceso es cada vez más usado en la industria.
  - El modelo de proceso V subraya la correspondencia entre actividades previas a la implementación de los programas (requerimientos, diseño general, diseño detallado) y las posteriores (prueba del sistema, pruebas integradas, pruebas unitarias) y la conveniencia de avanzar en la construcción de los planes de prueba tan pronto como sea posible como forma de validación de requerimientos y diseño y para adelantar tareas referidas a las pruebas.

8. Un proceso de diseño implica el desarrollo de diferentes vistas del sistema, y además:
- Implica obtener el OK de los usuarios del sistema para considerarlos válidos.
  - Se requiere esfuerzo no despreciable para desarrollar y mantener estas vistas.
  - Para desarrollos de grandes sistemas son un mecanismo de comunicación importante.
  - Para pequeños sistemas esto puede no ser rentable.

Seleccione la opción correcta:

- La afirmación (I) es correcta. El resto de las afirmaciones son incorrectas.
- Las afirmaciones (I) y (II) son correctas. El resto de las afirmaciones son incorrectas.
- Las afirmaciones (II) y (IV) son correctas. El resto de las afirmaciones son incorrectas.
- La afirmación (IV) es correcta. (Del resto de las afirmaciones esta respuesta no asegura nada)

9. Con respecto a los equipos de trabajo en un proyecto de software:

- Cuanto mayor sea el número de integrantes de un equipo de proyecto, la cantidad de líneas de comunicación será también mayor. Esto no constituye un riesgo para el proyecto, ya que si aumenta la cantidad de personas trabajando para un proyecto, existe una mayor probabilidad de que el proyecto sea exitoso.
- Es una ventaja para el equipo de proyecto que sus integrantes tengan el mismo perfil, habilidades y conocimientos, ya que esto incrementa la cohesión del grupo de trabajo.
- Conformar un equipo de proyecto adecuado es la clave de éxito de cualquier proyecto, independientemente de otros factores (tiempo, costo, calidad) que puedan afectarlo.
- Ninguna de las anteriores es correcta.

10. Dadas las siguientes afirmaciones respecto a pruebas de software:

- Los objetivos de la verificación son: mostrar que el software cumple con los requerimientos especificados y descubrir situaciones dónde el comportamiento sea incorrecto.
- Las revisiones e inspecciones son técnicas dinámicas de verificación y validación.
- Los defectos de interfaz de usuario son los más costosos de remover, sobre todo si se detectan durante el uso del sistema.
- El objetivo principal de las revisiones e inspecciones es encontrar defectos en la Arquitectura de software.

¿Cuál de las siguientes opciones es correcta?

- Las afirmaciones I) y II) son correctas y el resto son incorrectas.
- La afirmación I es correcta y el resto son incorrectas.
- Las afirmaciones III) y IV) son correctas y el resto incorrectas.
- Las afirmaciones I), III) y IV) son correctas.

11. Una empresa de construcción de software que aspira a certificarse Nivel 3 de CMMI en la categoría de ingeniería (representación continua) tiene:

- Que cumplir con todos las metas específicas y genéricas de las áreas de proceso de Ingeniería y de Soporte.
- Que cumplir con las metas específicas y genéricas de las áreas en las que aspira a certificarse hasta el nivel deseado.
- Sus procesos definidos, implementados y gestionados para toda la organización.
- Algunos procesos a Nivel 1, otros a Nivel 2, pero los más importantes a Nivel 3.



12. Respecto a la ingeniería de software:

- I. Es una disciplina que ha cambiado mucho durante los últimos veinte años, en consonancia con la creciente importancia del software en la sociedad actual y resulta esperable que esta disciplina continúe cambiando todavía más en los próximos años.
- II. Guarda relación con el área de Gestión de Recursos Humanos de Administración General, ya que la construcción de software es una rama industrial que hace un uso muy intensivo de la mano de obra. Asimismo, en tareas creativas, la motivación juega un papel muy relevante.
- III. Está relacionada con la ingeniería de sistemas, dado que en general se construye software como componente de un sistema compuesto al menos por software, hardware sobre el que ejecuta ese software y personas que lo utilizan, y se evalúa el desempeño del software en función del desempeño del sistema del que forma parte.
- IV. Los atributos de calidad internos, tales como la legibilidad del código, solo están disponibles para ser evaluados una vez que el software ya está construido y funciona lo suficiente como para mostrar su comportamiento.
- V. La creciente importancia del software en el funcionamiento de la sociedad en los más diversos aspectos ha estado acompañada por la creciente importancia de los temas legales, normativos, reglamentarios y éticos relacionados con su producción y utilización.
- VI. Persigue, entre otros objetivos, mejoras en la productividad y calidad en la producción de software.

Son correctas las respuestas:

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

13. Acerca de la trazabilidad de los requisitos:

- a) Para poder analizar las razones e impactos de una solicitud de cambio en los requisitos es necesario mantener únicamente las relaciones entre cada requisito y los componentes (de diseño y código) que lo implementan.
- b) Si la implementación de un cambio en los requisitos no representa cambios en el diseño, entonces no es necesario actualizar el documento de requisitos.
- c) Para poder analizar las razones e impactos de una solicitud de cambio es necesario mantener las relaciones entre los requisitos y el origen del mismo, además de las relaciones de dependencia entre requisitos, y las relaciones entre los requisitos y los componentes (de diseño y código) que lo implementan.
- d) Es algo deseable, pero imposible de realizar debido a que los requisitos cambian muy rápido y durante todo el ciclo de desarrollo.

14. Las técnicas estáticas en comparación con las dinámicas:

- I. Son más efectivas en la detección temprana de defectos que las dinámicas.
- II. Sirven para verificar cualquier producto (requerimientos, diseño, código, casos de prueba, etc) mientras que las dinámicas necesitan "algo" (unidad, sistema, modelo, etc.) ejecutable.
- III. No está sujeto a los errores de nuestro razonamiento mientras que las dinámicas sí.
- IV. Se basan en el software ejecutable mientras que las dinámicas se basan en modelos utilizados por las técnicas de prueba para derivar los casos de prueba.

¿Cuál de las siguientes opciones es correcta?

- a) Todas son correctas.
- b) I) y II) son correctas.
- c) II) y III) son correctas.
- d) III) y IV) son correctas.

15. ¿En cuál de las siguientes situaciones sería necesario revisar y (quizás) modificar la planificación de un proyecto de desarrollo de software?

- I. El cliente incluye un nuevo requerimiento.
- II. Un tester abandona el proyecto.
- III. Se modifica un objetivo de negocio que se pretendía alcanzar con el proyecto
- IV. Ingresar al proyecto un nuevo desarrollador con experiencia en la tecnología utilizada.
- V. El proceso es iterativo incremental y comienza una nueva iteración.

¿Cuál de las siguientes opciones es correcta?

- a) Se cumplen I), II) y IV)
- b) Se cumplen I), II) III) y IV)
- c) Todas son correctas.
- d) Ninguna es correcta.

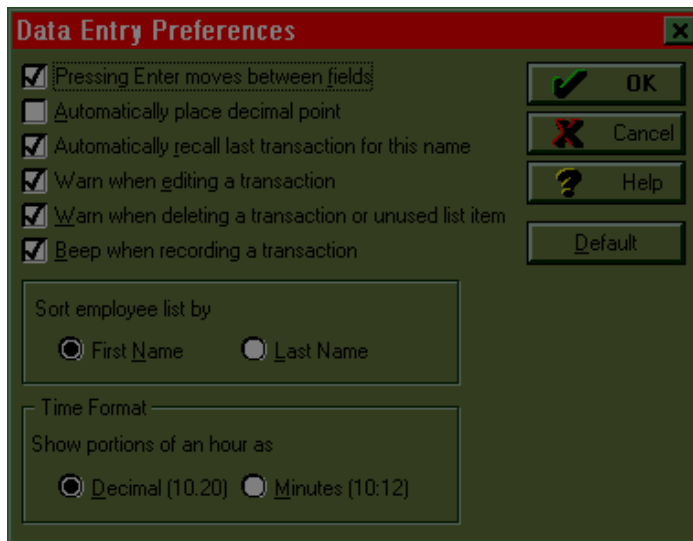


16. El cuestionario o encuesta se utiliza principalmente para:
- Evitar omisiones de requisitos.
  - Obtener datos estadísticos sobre preferencias.
  - Entender el negocio.
  - Validar asunciones.

Las opciones correctas son:

- I) y IV)
- I) y III)
- II) y IV)
- I), II), III) y IV)

17. Dada una funcionalidad que permite cambiar las preferencias de entrada de datos de una aplicación, que tiene la siguiente interfaz, donde el botón "Help" despliega una ventana de ayuda; el botón "Default" cambia las opciones elegidas a una selección predefinida; el botón "OK" graba las preferencias seleccionadas; y el botón "Cancel" cancela la operación.



La contribución de las transacciones debería contarse de la siguiente manera:

- una EI con 1 FTR y 9 DET /
- una EI con 1 FTR y 5 DET
- una EI con 1 FTR y 9 DET más una EQ con 1 FTR distinto y 8 DET
- una EI con 1 FTR y 12 DET más una E1 con 0 FTR y 8 DET

18. Los requisitos no funcionales incluyen:

- Restricciones que especifican el comportamiento del producto, así como las restricciones que se derivan de factores externos al sistema.
- Restricciones que se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.
- Declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares.
- (a) y (b) son correctas.

19. Indique cuál de las opciones siguientes **no** es un criterio de terminación de las pruebas de software según lo visto en el curso:

- Basado en el control de versiones del software. Cuando se pasa de una versión del software a una nueva versión de la línea base sin encontrar defectos se deben detener las pruebas.
- Basado en el diseño de casos de prueba. Se establece un criterio para diseñar los casos de prueba. Cuando los casos de prueba cumplen el criterio y además todos esos casos ejecutan sin fallas se terminan las pruebas.
- Basado en la cantidad de defectos remanentes. Se terminan las pruebas cuando queda cierta cantidad de defectos remanentes por Kloc. Para poder utilizar el criterio se utilizan técnicas que permiten estimar la cantidad de defectos remanentes.
- Ninguna de las anteriores.

20. Sobre las pruebas de software:
- I. Mediante las pruebas de software se puede comprobar la ausencia de defectos de un sistema software.
  - II. Durante las pruebas algunos defectos pueden enmascarar a otros defectos. Esta situación no sucede en la Inspección de software.
  - III. Los avances tecnológicos y de la ciencia permiten la realización de pruebas automatizadas. Si un desarrollo de software comienza de cero utilizando herramientas de vanguardia todas las pruebas de software durante el desarrollo serían automatizadas y las pruebas manuales no tendrían ningún sentido.
  - IV. Las pruebas unitarias deben ser automatizadas y garantizar el cubrimiento de sentencias.
  - V. Partición en clases de equivalencia es una técnica de prueba de caja negra. Esta técnica busca particionar el dominio de entrada del software bajo prueba. Cada partición tiene el supuesto de que los elementos de la partición tienen características similares por la que se procesan de la misma manera.
  - VI. El cubrimiento de código no tiene sentido en software desarrollado con Java.

¿Cuál de las siguientes opciones es correcta?

- a) II) y IV) son correctas.
- b) I), II) y V) son correctas.
- c) III), IV) y VI) son correctas.
- d) II) y V) son correctas.

---

21. Con respecto a la validación de requisitos:

- I. Es el proceso de verificar que los requerimientos especificados realmente definen el sistema que el cliente quiere.
- II. Si se encuentra un error durante la validación de los requisitos, es recomendable que sea corregido en las etapas siguientes (diseño o implementación). Esto disminuye el retrabajo, ya se corrigen en conjunto con los errores detectados también en esas etapas.
- III. Según el libro de referencia del curso, existen diferentes tipos de verificaciones que pueden hacerse sobre los requisitos. Algunos de ellos son: consistencia, completitud, verificabilidad y usabilidad.
- IV. El prototipado y la generación de casos de pruebas son algunas de las técnicas de validación posible.
- V. En general, durante la validación de requisitos se detectan y corrigen todos (o la mayoría) de los problemas existentes. La importancia de la validación está en que en las siguientes etapas del desarrollo no se detectarán más errores en los requisitos.

Son correctas las opciones:

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

---

22. Con respecto a la gestión de riesgos

- I. Sommerville clasifica los riesgos en riesgos del proyecto, riesgos del producto y riesgos del negocio. Estas categorías de riesgos pueden superponerse. Por ejemplo, el riesgo asociado a los cambios en los requisitos podría ser un riesgo tanto del proyecto como del producto.
- II. El proceso de gestión de riesgos consta de 4 pasos: identificación, planificación, monitoreo y evacuación de los riesgos.
- III. Cada uno de los riesgos debe ser evaluado periódicamente para decidir si el riesgo se está tornando más o menos probable, y también para ver si los planes de mitigación y contingencia han sido exitosos.
- IV. El proceso de gestión de riesgos es un proceso iterativo que continua a lo largo de todo el proyecto.

Las siguientes afirmaciones son verdaderas:

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

---

23. Respecto de la técnica de modelado de casos de uso:

- a) Es la mejor técnica de análisis de requisitos funcionales.
- b) Es la única técnica que sirve para modelar requisitos funcionales.
- c) Especifica qué es lo que el sistema debe hacer y cómo debe ser implementado.

- d) Tiene la limitación de que no es útil para representar requisitos no funcionales.
-



24. Entre las cualidades más relevantes para un software que cumpla la función de sistema operativo en un celular inteligente están:
- Adecuación al uso, Interoperabilidad, Tiempo de respuesta, Interfaz de usuario atractiva.
  - Interfaz de usuario atractiva, Modularidad, Facilidad de ser probado, Facilidad de mantenimiento.
  - Interoperabilidad, Disponibilidad, Calidad de la documentación técnica, Seguridad de la integridad física (safety).
  - Adecuación al uso, Modularidad, Legibilidad del código, Calidad de la documentación técnica.

25. En relación a la generación y ejecución de casos de prueba:
- Siempre es mejor utilizar más tiempo en el diseño de casos de prueba que en la ejecución de las pruebas.
  - Cuando se utilizan técnicas exploratorias no es necesario planificar las pruebas ni gestionarlas.
  - Siempre que sea posible, lo más riesgoso e importante para el cliente y/o para el negocio debería ser probado primero.
  - Siempre es mejor utilizar la técnica de cubrimiento de condiciones que cubrimiento de sentencias.

26. 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, no se identifica un subconjunto propio de funcionalidad que pueda resultar útil a 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

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 igualmente 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.

- A1, B1, C1
- A4, B2(1), C3(2(1))
- A2(3(1)), B3(1), C1
- A3(2(1)), B2(1), C4

27. Las normas de calidad de un producto de software ISO/IEC 9126 o la más reciente 25000:
- Describen características y atributos de un producto de software que se pueden medir con precisión.
  - Establecen criterios para definir los requisitos de calidad de un producto de software.
  - a) y se utilizan para evaluar un producto de software para su aceptación.
  - b) y definen tres vistas diferenciadas en el estudio de la calidad de un producto: interna, externa y en uso.

28. Los procesos CBSE dan soporte a la ingeniería de software basada en componentes, además:
- El proceso de desarrollo que sigue no está orientado a la reutilización.
  - Utiliza estándares para ayudar a la integración de componentes y un proceso de Certificación de Componentes para chequear cada componente y certificar que cumple con la especificación.
  - Se usa normalmente un proceso de desarrollo orientado a la reutilización. Sin embargo, en el desarrollo basado en componentes no se cuenta con un proceso para la gestión de los componentes por lo que la gestión de almacenamiento, y catalogación debe realizarse ad-hoc.
  - Una de las ventajas, es que como cada componente es una caja negra, se pueden usar los componentes de forma totalmente segura.

29. Se entiende por configuración:
- Elementos que componen la información producida como parte del proceso de ingeniería de software.
  - a), como ser código, documentos y datos.
  - a). Estos elementos pueden colocarse bajo línea base, lo que implica que no podrán ser modificados.

d) Los archivos que se encuentran en la línea base vigente.

---

30. Respecto al mantenimiento :

- I. Constituye una buena práctica adoptar una política de versiones de forma que los cambios se introduzcan en “paquetes” de cambios. Esto permite realizar una prueba del sistema a un costo razonable, ya que este se distribuye entre todos los componentes del paquete. Si los cambios se introducen de a uno, o bien no se lleva a cabo una adecuada prueba de regresión del sistema o el esfuerzo y costo asociado sería muy grande.
- II. La Gestión de la Configuración es de máxima importancia durante el desarrollo, no así durante el mantenimiento.
- III. Los gerentes suelen asignar mayor importancia al desarrollo que al mantenimiento lo que tiende a traducirse en un elemento desmotivador del personal asignado al mantenimiento.
- IV. El análisis de impacto de un cambio es una técnica que normalmente consume parte considerable del esfuerzo de mantenimiento. Herramientas que generan referencias cruzadas pueden asistir en este análisis.
- V. El contar con información de la historia de esfuerzo de mantenimiento correctivo por componente permite evaluar la conveniencia de re-escribir y/o re-diseñar en lugar de seguir realizando correcciones.
- VI. La información que se obtiene durante el mantenimiento de software permite evaluar la calidad del producto y de sus componentes, de forma análoga a lo que ocurre con la información que surge del testing durante el desarrollo

Son correctas las respuestas:

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