

- Cada pregunta de la **parte múltiple opción** contestada correctamente tiene un valor de **3,33** puntos. Cada pregunta incorrecta de la múltiple opción resta **1,11** puntos. Esta parte consta de 30 preguntas por lo que vale en total **100** puntos.
- Se aprueba con el 70% del puntaje.

Parte 1 Múltiple Opción

- 1) Respecto a la verificación y validación del software
 - a. El software falla cuando no hace lo requerido o hace algo que no debería.
 - b. a) y uno de los objetivos principales de la verificación y validación es descubrir defectos para luego poder corregirlos
 - c. b) y no es un objetivo de la verificación el evaluar la calidad de los productos
 - d. Ninguna de las anteriores

- 2) En el criterio de cubrimiento de condición múltiple el Conjunto de Casos de Prueba debe asegurar que:
 - a. Todas las combinaciones posibles de resultados de condición dentro de una decisión se ejecuten al menos una vez.
 - b. Cada condición dentro de una decisión debe tomar al menos una vez el valor TRUE o el valor FALSE en el conjunto de casos de prueba
 - c. Se ejecuta al menos una vez cada instrucción del código y no necesariamente asegura a)
 - d. Ninguna de las anteriores.

- 3) Seleccione la que no es una estrategia de integración
 - a. Big Bang
 - b. Eliminación redundante
 - c. Top-down
 - d. Sándwich

- 4) En el proceso de Verificación y Validación las técnicas estáticas de verificación
 - a. son efectivas en la detección temprana de defectos.
 - b. (a) y sirven para verificar no solo el código sino también los requerimientos y el diseño.
 - c. (b) y son muy utilizadas como técnicas de validación.
 - d. (b) y en diversos estudios empíricos muestran una alta efectividad en la detección de defectos (más de 60%)

- 5)
 - a. La verificación es un proceso costoso en comparación con otros procesos del desarrollo de software.
 - b. (a) y por esto es conveniente no realizar ningún tipo de verificación durante el desarrollo
 - c. Las pruebas a nivel de sistema se deben detener cuando no se detecten más fallas.
 - d. Ninguna de las anteriores.

6) Acerca de la ingeniería de software:

- a. Es una disciplina que es esencialmente idéntica a la programación de computadoras.
- b. Existe una fuerte relación entre la ingeniería de software y la ingeniería de sistemas, ya que el software que construimos va a formar parte de un sistema conformado por el software en sí, el hardware sobre el que corre y las personas que lo utilizan.
- c. Tiene como principal objetivo construir software de calidad excelente y cero defecto.
- d. Es actualmente reconocida como una disciplina profesional, de forma análoga a la medicina, abogacía, ingeniería civil, con exigencias y controles para su ejercicio comparables a los de estas profesiones.

7) Entre los atributos de calidad más relevantes para un producto de software para la compra por Internet mediante tarjeta de crédito de artículos de supermercado están:

- a. tiempo de respuesta, adecuación al uso, consumo de recursos, seguridad de la información (security)
- b. tiempo de respuesta, adecuación al uso, consumo de recursos, seguridad de la integridad física (safety)
- c. adecuación al uso, facilidad de mantenimiento, amigabilidad de la interfaz de usuario, modularidad
- d. seguridad de la información (security), facilidad de mantenimiento, facilidad de verificación, facilidad de adaptación

8) Respecto a modelos de proceso de software:

- a. Una de las desventajas de un proceso en fases con liberaciones parciales en producción es que no permite acortar el plazo para poder contar con resultados de utilidad, lo que cada vez es más requerido para reducir el plazo para salir al mercado (time-to market).
- b. El proceso en cascada al día de hoy prácticamente ha dejado de ser utilizado en la industria, habiendo sido sustituido por otros modelos como el proceso en fases.
- c. Un proyecto se va a encarar siguiendo un proceso en fases con liberaciones parciales en producción. Se ha definido que el proyecto se va a desarrollar en 3 fases. Podría resultar adecuado adoptar una estrategia de liberación paulatina para las 2 primeras fases y big-bang para la última.
- d. c) y un modelo de proceso se define normalmente con uno o ambos de los objetivos siguientes:
 - 1. Describir de qué manera se hacen las cosas en una organización (modelo descriptivo). Esto permite razonar sobre el modelo, lo que resulta mucho más fácil que razonar sobre la realidad.
 - 2. Prescribir de qué manera se deben hacer las cosas en una organización (modelo prescriptivo).

9) Dados los siguientes proyectos y modelos de proceso.

Proyecto:

A - de alto riesgo, muy complejo, alcance no del todo definido, 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 Liberaciones Parciales Internas

3 - En Fases con Liberaciones Parciales en Producción

4 - 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 igualmente adecuado.

- a. A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4
- b. A2, B2, B3, C4
- c. A1, B1, C3, C4
- d. A4, A3, B2, B3, C1

10) Respecto a la liberación del software:

- a. Una desventaja de la estrategia big-bang en el caso de un sistema que sustituye a un sistema anterior está relacionada con el esfuerzo que requiere tener al sistema nuevo y viejo operativos al mismo tiempo.
 - b. Al definir el alcance de un proyecto debe quedar claro si incluye o no tareas de la liberación del software tales como capacitación, entrenamiento a los distintos tipos de usuarios, documentación para usuarios, instalación, soporte y solución de problemas en etapas iniciales de la implantación, conversión de datos de un sistema anterior al sistema nuevo.
 - c. b) y son posibles distintos tipos de estrategias de implantación paulatina en una organización. Por ejemplo por tramos de usuarios o clientes, por tramos de productos o servicios, por área geográfica.
 - d. Los sistemas legados (construidos hace muchos años, en general con tecnologías hoy en día obsoletas) que tienen costos de mantenimiento muy altos y prestaciones insuficientes, son en general fáciles de sustituir por sistemas nuevos si estos están implementados con tecnologías modernas.
-

11) Respecto al mantenimiento de software:

- a. Normalmente se presenta un conflicto entre las necesidades de corto plazo y las de largo plazo. A corto plazo importa tener una solución adecuada y a la vez rápida. A largo plazo importa mantener costos de mantenimiento reducidos.
- b. a) y si no se toman medidas al respecto, con las sucesivas modificaciones, los programas “envejecen”, pierden atributos de calidad y en particular se vuelven más difíciles de entender y mantener.
- c. **b) y el mantenimiento se clasifica en las categorías siguientes de acuerdo al objetivo de cada modificación:**
Correctivo – para eliminar defectos encontrados en el software, por ejemplo un cálculo con un redondeo erróneo
Adaptativo – para adaptarlo a cambios en el ambiente del software, por ejemplo un cambio resultado de una modificación legal o reglamentaria
Perfectivo – para mejorar algún aspecto ya presente en el software, por ejemplo para disminuir el consumo de recursos o para que la experiencia del usuario resulte más satisfactoria
Preventivo – para prever fallas futuras, por ejemplo reescribir un componente que históricamente ha demostrado ser propenso a fallar.
- d. c) y normalmente la gerencia de las organizaciones de desarrollo de software presta especial atención al mantenimiento de software y los equipos de mantenimiento suelen estar particularmente motivados por su trabajo.

12) Respecto a los cambios en un producto de software:

- a. Debiera definirse un Comité de Control de Cambios, con la responsabilidad de controlar los cambios que se solicitan y aplican sobre ese software. Este comité debiera estar compuesto por un subconjunto del equipo de mantenimiento que cuente con una visión global del software a mantener.
- b. **Debiera definirse un procedimiento de Control de Cambios que asegure que se cumplan los pasos necesarios para la aplicación de cambios, entre ellos su registro, de forma que se sepa para cada paso: cuándo, quién, por qué, qué cambió y aprobó.**
- c. Para el caso de que se requiera introducir un cambio debido a una emergencia (por ejemplo un defecto de programa que impide el uso de alguna funcionalidad crítica), suele resultar adecuado aplicar el procedimiento normal de cambios, con la condición de que los involucrados en cada paso del procedimiento lo hagan lo más rápido posible.
- d. Respecto a la aplicación de cambios en Producción, conviene en general adoptar la política de aplicar cada cambio tan pronto como sea posible, ya que esto facilita la realización de pruebas de regresión adecuadas.

13) Durante la elaboración del diseño de software

- a. **Se debe buscar preservar la arquitectura.**
- b. a) Es altamente recomendable que distintos grupos de componentes no sean realizados por distintas personas.
- c. a) En caso de no lograr este objetivo se deberá modificar la arquitectura, ya que el diseño es la etapa más importante de la construcción del software.
- d. a) Es recomendable evaluar distintas alternativas de diseño, eligiendo luego la que resulte de mayor eficiencia.

-
- 14) El acoplamiento:
- Es un indicador sobre la dependencia entre módulos.
 - a) Se dice que dos módulos están "altamente acoplados" si la dependencia entre estos puede ser despreciable.
 - El alto acoplamiento facilita la comprensión y mantenimiento de los módulos del sistema.
 - a) El acoplamiento de un módulo A que usa a un módulo B, depende de cuánta información se necesita conocer sobre el módulo B para entender y modificar el módulo A y también de qué tan compleja es esta información
-
- 15) La cohesión:
- Se focaliza en conocer porqué los elementos de un módulo están juntos en ese módulo.
 - a) El objetivo es tener en un mismo módulo elementos que están fuertemente vinculados.
 - c) La cohesión "por interacción" se presenta cuando métodos de una clase invocan a métodos de otra clase
 - Ninguna de las anteriores.
-
- 16) Son ventajas de diseñar y documentar la arquitectura de software:
- Poder comunicarla entre los stakeholders.
 - a) Poder tomar decisiones tempranas de diseño.
 - b) Lograr importantes niveles de reuso.
 - c) Para priorizar los requerimientos de forma temprana
-
- 17) Las características de calidad de un producto de software y los atributos asociados:
- Varían según los distintos tipos de software
 - Son iguales para todos los productos de software
 - Dependen del criterio de los usuarios y clientes
 - Ninguna de las anteriores
-
- 18) El modelo recomendado por Collier, DeMarco y Fearey para el análisis post-mortem de un proyecto incluye:
- Recopilar información objetiva del proyecto sin comprometer su confidencialidad
 - Comenzar poco antes de culminar el proyecto
 - a) y mantener reuniones para completar la información faltante y reflexionar sobre todos los datos disponibles
 - Ninguna de las anteriores
-
- 19) Se denomina desempeño de un proceso a:
- La estimación de los resultados que se obtendrán cuando se ejecute el proceso
 - La medida (indicadores) de los resultados reales obtenidos luego de su ejecución
 - Cuán difundido está el proceso en la organización
 - b) y se relaciona con las predicciones a través de la definición de su capacidad
-

20) Con respecto a la Gestión de la Configuración

- a. Los antipatrones mencionados en clase corresponden a situaciones que ocurren habitualmente en dicha disciplina y deben evitarse.
- b. a) y es posible que no todas las situaciones planteadas se puedan evitar utilizando herramientas para gestionar la configuración
- c. No es posible realizar la gestión de la configuración en un proyecto sin utilizar herramientas de software que apoyen las actividades involucradas.
- d. Ninguna de las anteriores

21) La Configuración del Software

- a. comprende los elementos que componen toda la información generada durante del proceso de ingeniería de software.
- b. a) y algunos ejemplos pueden ser: especificación de requerimientos, conjuntos de casos de prueba, reportes de defectos, manuales de usuarios.
- c. b) pero esto no quiere decir que desde que se crea un elemento de configuración es necesario controlar los cambios mediante procedimientos formales.
- d. c) y un elemento de configuración se convierte en línea base cuando se ha revisado formalmente, inclusive si no se ha llegado a un acuerdo.

22) Si el gerente de un proyecto percibe que no se llegará al plazo comprometido, puede intentar:

- a) negociar un alcance reducido
- b) negociar un plazo extendido
- c) contratar más personal
- d) todas las anteriores

23) El término stakeholders:

- a. hace referencia a las personas que puedan ver sus intereses afectados por la ejecución o el éxito el proyecto
- b. a) y puede indicar también organizaciones que puedan ver sus intereses afectados por la ejecución o el éxito el proyecto
- c. solo indica aquellas personas que están activamente involucradas en el proyecto
- d. incluye a todos los gerentes de las demás divisiones de la organización.

24) ¿Cuáles de las siguientes cualidades es la más necesaria para un gerente de proyecto?

- a. sólida experiencia en tecnologías
- b. ser una persona querida por el equipo de desarrollo
- c. conocer sobre distintas técnicas para relevar requerimientos de software
- d. manejar los fundamentos y técnicas de gestión de proyectos

25) Una vez realizado el plan de una iteración

- a. sirve de línea base contra la que contrastar la realización
- b. no debe ser ajustado hasta la nueva iteración
- c. alcanza con verificar el perfil de uso de recursos de los desarrolladores para saber si es factible
- d. todas las anteriores

-
- 26) La disciplina de Ingeniería de Requisitos es importante porque:
- es en el relevamiento de requisitos donde se producen la mayor cantidad de defectos
 - un defecto del software debido a un error en el momento del relevamiento de requisitos es normalmente mucho más costoso de arreglar que un defecto introducido durante la implementación
 - los sistemas tienen siempre requisitos inciertos y cambiantes a lo largo del proyecto
 - todas las anteriores
-
- 27) En cuanto a los requisitos funcionales y no funcionales:
- Los requisitos funcionales se describen de forma cuantitativa y los no funcionales de forma cualitativa
 - La verificación de los requisitos no funcionales es subjetiva
 - Los requisitos no funcionales describen restricciones que limitan las elecciones para construir una solución
 - Todas las anteriores
-
- 28) Las etapas de obtención y de análisis de los requisitos definidas en el curso incluyen:
- el relevamiento de los requisitos
 - a) y la clasificación y organización de los requisitos
 - b) y la priorización de los requisitos
 - c) y la validación de los requisitos
-
- 29) Una entrevista:
- debe apegarse a un conjunto pre-establecido de preguntas
 - no debería ser improvisada
 - es costosa y puede ser sustituida por una encuesta, si hay muchas personas a ser entrevistadas, con resultados equivalentes
 - no es una técnica conveniente de utilizar con usuarios
-
- 30) En una tormenta de ideas:
- se van discutiendo ideas a medida que se van proponiendo una a una
 - no se admite la censura en la fase de reducción
 - se pueden generar ideas por asociación con ideas propuestas por otros
 - en la fase de reducción, si hay desacuerdo, la idea se elimina