

- Cada pregunta **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

- Respecto al diseño de la arquitectura de un sistema de software:
 - Es una actividad que se lleva a cabo en etapas tempranas del proceso de desarrollo.
 - Comprende la identificación de los principales componentes del sistema y sus relaciones.
 - Tener una visión de alto nivel de la arquitectura es útil para la comunicación con los stakeholders del sistema y la planificación de proyectos.
 - En el modelo 4 + 1 la vista lógica presenta abstracciones claves del sistema como objetos o clases de objetos.

¿Cuál de las siguientes opciones es correcta?

 - Sólo I y II son correctas.
 - Sólo I, II y III son correctas.
 - Sólo I, II y IV son correctas.
 - Todas son correctas.**

- Sobre el modelo CMMI:
 - Las prácticas específicas y genéricas asociadas a las metas son obligatorias.
 - Cada área de proceso tiene metas específicas y genéricas que es obligatorio cumplir.
 - (b) y se espera que se apliquen las prácticas específicas y genéricas asociadas.**
 - Ninguna de las anteriores.

- La arquitectura en capas es un estilo arquitectónico con las siguientes características:
 - Cada capa provee servicios a la capa inmediatamente inferior.
 - No es apropiada cuando se tiene varios equipos de desarrollo.
 - Permite cambiar una capa entera (manteniendo los servicios provistos) sin necesidad de que el resto de las capas se vean alteradas.
 - (c) Como debilidad, este tipo de arquitectura puede penalizar la performance de la solución en determinados escenarios.**

- Los sistemas legados:
 - Son sistemas que dan soporte al funcionamiento de una organización y fueron desarrollados por personal de una empresa contratada.
 - Son sistemas que fueron desarrollados hace tiempo, a menudo con tecnologías hoy obsoletas, lo que dificulta su mantenimiento.**
 - Suelen tener costos de mantenimiento muy bajos ya que normalmente los costos de mantenimiento del software decrecen con el paso del tiempo.
 - A menudo se trata de sistemas muy relevantes para el funcionamiento de la organización y en estos casos no se explica que no hayan sido sustituidos por sistemas que tengan un menor costo de mantenimiento.

- Con respecto a la interfaz de usuario:
 - Debe utilizar términos y conceptos obtenidos de la experiencia de las personas que más utilizan (utilizarán) el sistema.**
 - No debe proveer mecanismos para permitir a los usuarios recuperarse de errores ya que no es responsabilidad de la interfaz.
 - Debe cumplir con el principio de "máxima representatividad".
 - (a) y (b) son verdaderas.

- La ventaja de usar el GQM (Goal Question Metric - Objetivo Interrogante Métrica) para la mejora de procesos consiste en que:
 - Separa conceptualmente los intereses de la organización (goal) de los cometidos específicos del proceso (question).
 - Proporciona una base para decidir qué información recolectar.
 - Sugiere que los datos obtenidos (métricas) se analicen de forma diferente según la pregunta que pretenden responder.
 - Todas las anteriores.**

7. Los patrones de diseño:
- Recogen soluciones probadas a problemas conocidos.
 - (a), y permiten trabajar sobre un "vocabulario común".
 - (b), y facilitan la comunicación con distintos stakeholders.**
 - (a), y deben incorporarse a la Gestión de la Configuración del proyecto.
-
8. En cuanto a la Gestión de la Configuración (SCM) se puede decir que:
- Abarca gestión de cambios, gestión de las versiones, y gestión de las liberaciones.**
 - No es necesaria cuando se trabaja con equipos en el mismo lugar ya que la comunicación continua basta para conocer el estado de las versiones y decidir acerca de los cambios.
 - No es necesaria luego de liberar un sistema, ya que una vez en uso no se deberán atender cambios.
 - La línea base es la colección de versiones de ítems de configuración (documentos, código, diagramas, datos de prueba) que se define justo antes de la liberación del sistema a los clientes o usuarios.
-
9. Respecto a los costos del mantenimiento:
- Considerando el ciclo completo de la vida del software estos suelen ser de un orden de magnitud comparable a los del costo de desarrollo.**
 - Los costos del mantenimiento suelen ser cada vez menores a medida que pasa el tiempo ya que suele resultar cada vez más fácil conseguir personal que se dedique al mantenimiento y este pasa a ser productivo de forma más rápida que si se incorporara a un equipo de desarrollo.
 - La sucesión de intervenciones sobre los productos de software tiende a volverlos cada vez más simples, lo que facilita su mantenimiento. En general, a medida que el software se mantiene, este resulta cada vez más fácil de mantener.
 - Tanto el análisis de impacto como las pruebas de regresión, tienen muy poca incidencia en el esfuerzo de mantenimiento y por lo tanto su incidencia en los costos de mantenimiento es muy poco significativa.
-
10. Respecto a la liberación de un sistema:
- La planificación de la liberación del sistema conviene que comience una vez que se cuenta con una arquitectura estable, capaz de soportar la totalidad de los requerimientos no funcionales para el producto.
 - En aquellos casos en que se requiere realizar un entrenamiento a los usuarios del sistema de forma previa a su liberación, es conveniente disponer de documentación que lo facilite.
 - La documentación del sistema conviene que sea lo más amplia posible. Todo esfuerzo que se dedique a la elaboración de documentación repercutirá reduciendo el esfuerzo necesario para la solución de problemas una vez que el sistema esté en uso.
 - El entrenamiento a los usuarios constituye una forma de validación del producto.
 - El entrenamiento a los usuarios conviene que se lleve a cabo poco antes de la fecha de puesta en producción para evitar pérdidas por olvido de lo aprendido.
 - Los usuarios que participen en la prueba del sistema son buenos candidatos tanto para brindar entrenamiento como para brindar soporte del sistema en producción.
- Son correctas:
- II, IV, V, VI.**
 - III, IV.
 - II, III, V, VI.
 - I, II, III, IV, V, VI.

-
11. ¿Cual de las siguientes afirmaciones es correcta?
- La mejora de los procesos es condición necesaria y suficiente para mejorar la calidad de los productos de software.
 - El enfoque ágil es más adecuado para abordar la mejora de procesos que apuntar a la madurez de la organización.
 - La capacidad de un proceso es la probabilidad de que la medida de su desempeño coincida con las estimaciones.**
 - Algunos atributos de calidad externos son: mantenibilidad, usabilidad, cantidad de líneas de código, cantidad de mensajes de error, entre otros.
-
12. El proceso de diseñar la arquitectura:
- Es un proceso rígido y preestablecido, donde no se debe tomar en cuenta cómo será evaluada la arquitectura, pues eso es competencia del responsable de calidad.
 - Considera únicamente los requerimientos funcionales y la distribución de los componentes.
 - Es un proceso creativo, donde se consideran los requerimientos funcionales, los requerimientos no funcionales, y se tiene en cuenta cómo será evaluada la arquitectura.**
 - Es un proceso que incluye decisiones de distribución del sistema y estilos arquitectónicos, considerando requerimientos funcionales y no funcionales, y donde no se debe tomar en cuenta cómo será evaluada la arquitectura, pues eso es competencia del responsable de calidad.

13. Dado un proyecto que incluye análisis, diseño, construcción, prueba y puesta en producción (liberación) del software, para sustituir el sistema que da soporte al funcionamiento de una organización por un sistema nuevo y al que hay que migrar información histórica del sistema viejo:
- La planificación de la migración del sistema viejo al nuevo debiera comenzar desde el inicio del proyecto, a los efectos de considerarla en el alcance del proyecto.
 - De forma previa a la puesta en producción se podría realizar un procesamiento en paralelo para probar con datos reales, evaluar el comportamiento del sistema con los datos migrados del sistema viejo y poner a prueba los procedimientos de la organización con el sistema nuevo.
 - Una estrategia de implantación big-bang permite reducir los riesgos asociados a problemas en los procedimientos de la organización.
 - Una estrategia de implantación paulatina permite manejar de mejor forma los riesgos asociados a problemas en los procedimientos de la organización.
 - Una estrategia de implantación big-bang evita los problemas derivados de la coexistencia en la organización de un sistema viejo y otro nuevo.
 - En la planificación de la puesta en funcionamiento del sistema nuevo conviene prever recursos para atender la solución de problemas de diverso tipo (software, instalación, servidores, entrenamiento, procedimientos, etc.).

Son correctas:

- I, II, IV, V, VI.
- II, III, V.
- I, II, III, IV
- I, IV, V, VI.

14. Con respecto al diseño de software:

- El diseño orientado a objetos facilita la reutilización.
- El principio de abierto-cerrado busca promover la definición de módulos abiertos para la modificación y cerrados para la extensión.
- El diseño basado en componentes puede ser visto como un nivel de abstracción superior al diseño orientado a objetos ya que se busca reusar colecciones de objetos.
- Por lo general cuanto más acoplados sean los módulos, más reusable serán los mismos.
- Las técnicas de modularidad y ocultamiento de la información ayudan a que el sistema sea más reusable.

Las siguientes afirmaciones son verdaderas:

- I,II,V
- I,II,IV
- I,III,V
- II,IV,V

15. En el mantenimiento de software:

- Resulta conveniente disponer de un ambiente completo (fuentes, ejecutable, datos) en el que poder llevar a cabo y probar las modificaciones, separado del ambiente del software en producción.
- Se debe disponer de un procedimiento para los cambios normales y otro para los cambios de emergencia.
- Para llevar a cabo la prueba del sistema, conviene disponer de un ambiente de prueba independiente de aquél en que se realizan las modificaciones a los programas.
- La gestión de la configuración resulta esencial durante el mantenimiento para asegurar:
 - que las modificaciones se realizan partiendo de las versiones adecuadas de los fuentes
 - que los ejecutables que se prueban son consistentes con los fuentes modificados y
 - que lo probado es lo que se pone en producción.
- Los cambios que se llevan a cabo debieran contar con la aprobación de un Comité de Control de Cambios que integre la visión del cliente y del equipo de desarrollo (o de mantenimiento).
- Resulta conveniente establecer una política de versiones de forma tal que los cambios se incorporan en paquetes de cambio en lugar de hacerlo de forma individual ya que esto permite distribuir el esfuerzo de la prueba del sistema entre varias intervenciones.

De las afirmaciones anteriores son correctas:

- I, II, III, IV, V, VI.
- I, III, IV, V, VI.
- II, III, IV, V.
- I, VI.

-
16. La línea base de un proyecto:
- Implica haber revisado todos los documentos generados durante el proyecto y que quieran ser liberados al cliente.
 - Puede cambiarse a través de los procedimientos de control de cambios.
 - Cambia a medida que se van generando nuevas versiones de los elementos que la componen.
 - (b) y un elemento de configuración se convierte en línea base cuando fue revisado y aprobado por el cliente.
-
17. La definición de la arquitectura del sistema:
- Implica involucrar a los usuarios finales para obtener su validación.
 - Se logra validar completamente una vez se realiza la liberación del sistema.
 - Es la actividad del proyecto que requiere mayor esfuerzo.
 - Ninguna de las opciones es correcta.
-
18. 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 30690 horas-persona. El 68% del esfuerzo corresponde a mantenimiento correctivo de 127 componentes de un total de 1437. El 78% de ese esfuerzo se concentró en 12 componentes. El 32% del esfuerzo de mantenimiento corresponde a mantenimiento adaptativo. No se realizó en el período ningún mantenimiento de tipo perfectivo ni preventivo.
- A partir de esta información sería conveniente considerar y evaluar con más detenimiento:
- Destinar esfuerzo al mantenimiento perfectivo.
 - Destinar esfuerzo a la reescritura de por lo menos alguno de los 12 componentes con mayor mantenimiento correctivo, con el objetivo de reducir de manera significativa la carga de mantenimiento correctivo.
 - Probar de forma exhaustiva los 127 componentes a los que se les realizó mantenimiento correctivo para detectar y corregir defectos antes de que estos aparezcan durante el uso del software en producción.
 - Someter a pruebas a los 1310 componentes que no tuvieron mantenimiento correctivo, a los efectos de detectar y corregir los defectos que puedan tener, antes de que estos aparezcan en el uso del software en producción.
-
19. Los estándares de calidad:
- Se construyen por consenso en base al conocimiento, experiencia y dificultades de los profesionales de determinada disciplina.
 - Se deben revisar periódicamente para acompañar los cambios tecnológicos.
 - Son recetas que se aplican al pie de la letra en cualquier contexto.
 - Un ejemplo de estándar mencionado en clase es la guía para el diseño e implementación de portales estatales
- Seleccione la opción correcta:
- Sólo I y II son correctas.
 - La opción II no es correcta.
 - Sólo I y III son correctas.
 - Sólo I, II, IV son correctas.
-
20. Respecto al mantenimiento de software:
- Los cambios se deben introducir de forma que se intente asegurar la continuidad del correcto funcionamiento del software en producción y del sistema que este conforma junto con el hardware en el que corre y las personas que lo utilizan, por los que solo se pueden poner en producción modificaciones que son correctas en el sentido de que cumplen con su especificación.
 - Existe un conflicto entre las necesidades de corto plazo (solucionar rápidamente los problemas, ya sean estos originados en defectos del software o en la necesidad de adaptarlo a cambios ambientales) y las necesidades de largo plazo de mantener costos de mantenimiento bajos y de preservar los atributos de calidad relevantes del producto.
 - El personal de mantenimiento de software normalmente se ve muy motivado por la importancia que suele asignarle la gerencia al mantenimiento respecto al desarrollo y por trabajar con tecnologías ya conocidas y probadas.
 - La calidad del software que está bajo mantenimiento suele ir mejorando de forma sostenida con el paso del tiempo, a medida que se van detectando defectos durante su uso y estos son corregidos.
-
21. La mejora de procesos de software tiene por objetivos principales:
- Incrementar la calidad de los productos de software.
 - a) y reducir los costos así como acelerar la liberación de productos al mercado.
 - Certificar los productos de software en determinado estándar aceptado internacionalmente.
 - Incrementar la motivación del equipo de desarrollo y la satisfacción del cliente.

22. Respecto a la arquitectura de software:
- a) No es una buena práctica adaptar estilos arquitectónicos a la necesidad del proyecto, las buenas prácticas deben tomarse tal como están definidas.
 - b) La arquitectura puede describirse desde diversos puntos de vista, por ejemplo: la vista de despliegue (o deploy) y la vista de componentes.
 - c) La arquitectura no puede verificarse ya que incluye requerimientos no funcionales, pero si puede validarse con el cliente.
 - d) Todas las respuestas son correctas.
-
23. ¿Qué atributos de calidad interna mediría si tuviera que evaluar la fiabilidad de un producto de software?
- a) Tolerancia a fallos y capacidad de recuperación.
 - b) a) y disponibilidad y facilidad de aprendizaje.
 - c) b) y madurez.
 - d) a) y disponibilidad.
-
24. Al diseñar la interfaz de usuario:
- a) Deben considerarse aspectos culturales de los usuarios finales.
 - b) Pueden generarse prototipos de las pantallas a desarrollar.
 - c) (b), y para lo cual debe aprenderse a utilizar programas específicos para esta función
 - d) Solo (a) y (b) son correctas.
-
25. Algunas métricas estáticas de un producto de software son:
- a) Número de líneas del código, profundidad de los anidamientos, número ciclomático
 - b) (a) y número de incidentes detectados en testing.
 - c) (a) y existe evidencia de que son aplicables en todos los contextos.
 - d) (a) Además, no existe suficiente evidencia de que apliquen a contextos como el de un ERP