

Instrucciones

- Cada pregunta múltiple opción respondida correctamente tiene un valor de 4,762 puntos.
- Cada pregunta múltiple opción respondida incorrectamente resta 1,587 puntos.
- Ante dos opciones correctas en una pregunta, se debe seleccionar la opción más completa.
- La evaluación es de carácter individual y la duración es de dos horas.
- El puntaje total es de 100 puntos.

1. Seleccione la respuesta correcta

- El incremento, de los últimos años, de la complejidad de los sistemas de software que se construyen es un factor más que influye en los defectos (severidad, cantidad, etc.) que los desarrolladores introducen en el software durante el desarrollo.
- La ingeniería de software está diseñada para dar soporte a la programación personal (individual), y este es su foco principal.
- Los problemas éticos que pueden surgir en el desarrollo de software (por el tipo de trabajo de un ingeniero de software) son menores en comparación a los que surgen en el resto de las ingenierías.
- Al desarrollar software, fallar en el uso de las técnicas y métodos de la ingeniería de software no influye en tener mayores costos, problemas de calidad o de mantenimiento. Estos problemas están más bien relacionados con aspectos de construcción de software a bajo nivel (codificación mayormente).

2. Seleccione la respuesta correcta

- Todos los procesos de desarrollo de software se pueden derivar del Capability Maturity Model Integration (CMM) como una instancia particular de ese modelo. Esto es porque el CMMI contiene la descripción de las actividades utilizadas en el desarrollo de software.
- La especificación de un proceso de desarrollo de software tiene que describir, entre otras cosas, las actividades a realizar, los roles, las fases en las que se divide el proyecto, las precedencias entre actividades, la forma de registrar el avance de los proyectos y registrar el esfuerzo, las herramientas de soporte al proceso, la forma de comunicación entre los integrantes del equipo de desarrollo. Si no contiene estas descripciones esenciales no se considera que sea un proceso de desarrollo de software completo.
- Los procesos que se basan fuertemente en planes realizan el plan completo del proyecto una única vez y hasta que no se termine el proyecto el plan no debe cambiar.
- Los procesos que se basan fuertemente en planes normalmente permiten modificar el plan a medida que avanza el proyecto. Es una forma de evitar que el plan sea obsoleto y no tenga ninguna utilidad ni para guiar qué tareas realizar ni para hacer un seguimiento del proyecto.

3. Seleccione la respuesta correcta

- Scrum no es un método iterativo e incremental
- Los métodos ágiles son adecuados cuando en un proyecto no está clara la totalidad de los requisitos y se requiere una planificación detallada a largo plazo.
- En los métodos ágiles se realizan las actividades de diseño de software que se creen necesarias para el mejor desarrollo del proyecto.
- En los métodos ágiles no se realizan actividades de diseño de software, ya que se hace refactoring cuando es necesario y esto se hace directamente a la hora de codificar/implementar.

4. Seleccione la respuesta correcta

- a) Los proyectos que usan procesos ágiles normalmente necesitan actividades de gestión de proyectos.
- b) (a) y normalmente esta gestión se realiza con un *burndown chart* pero sin realizar gestión de riesgos, gestión de costos o gestión de la calidad. Esta última, la gestión de la calidad, se desplaza para dar lugar a las retrospectivas del equipo y a la validación en cada iteración con el cliente (o *product owner*).
- c) Las historias de usuario, cuando se utiliza Scrum en conjunto con alguna práctica o técnica de XP, deben estar todas escritas, al menos a alto nivel, en los primeros Sprints para poder estimar el cronograma total del proyecto.
- d) Siempre se utilizan historias de usuario o casos de uso para especificar los requisitos cuando se utilizan métodos ágiles. Nunca se utiliza otro tipo de descripción de los requisitos porque disminuiría la velocidad del equipo dada la poca agilidad que se tiene al cambiar grandes documentos de texto o diseños de interfaces gráficas.

5. Seleccione la respuesta correcta

- a) La construcción de software como disciplina, según propone el SWEBOK, debe desarrollar primero los casos de prueba y luego el código. Esto se llama probar primero (*test first*) y es una forma liviana de desarrollo basado en pruebas (*test driven development*).
- b) La disciplina de construcción de software abarca, entre otras cosas, el diseño detallado de software y las pruebas unitarias.
- c) En la industria es común que los equipos de desarrollo realicen diseños detallados. Luego, esos diseños son tomados por una herramienta basada en inteligencia artificial que genera el código del sistema. A esto se le llama construcción basada en modelos.
- d) La construcción de software moderna se basa fuertemente en codificar y corregir. Es decir, se busca realizar una codificación rápida, para culminar lo antes posible la historia de usuario que un desarrollador tenga asignada, para luego pasar a pruebas, provocar fallas, detectar defectos y luego corregirlos. Esta es la forma que ha mostrado, durante la construcción, ser la más efectiva para cumplir con los plazos de los proyectos.

6. Se le ha encomendado obtener los requisitos para un sistema nuevo que automatiza procesos en una pequeña empresa que realiza muchas tareas complejas y difíciles de explicar, algunas interconectadas entre sí. Además, los usuarios no saben articular adecuadamente lo que hacen. ¿Qué técnica le parecería más adecuada para obtener los requisitos?

- a) etnografía
- b) investigar antecedentes
- c) casos de uso
- d) encuestas

7. En cuanto a los prototipos:

- a) Es fundamental establecer los objetivos que se buscan con cada prototipo previamente a su construcción.
- b) Es siempre mejor hacer un prototipo evolutivo, ya que eso ahorra trabajo posterior de desarrollo.
- c) Se debe procurar, con un único prototipo general, cubrir varios objetivos: relevar y validar requisitos, eliminar riesgos técnicos, aprender tecnologías, probar diseños y arquitecturas posibles y medir la productividad del equipo.
- d) Las metodologías ágiles requieren que todos los prototipos que se hagan sean evolutivos.

8. Respecto al rol de *product owner* en Scrum:
- a) Cualquier *stakeholder* puede cambiar directamente el contenido o la priorización del *backlog* del producto.
 - b) Los *stakeholders* que quieran cambiar el *backlog* del producto podrán hacerlo solo a través del *product owner*, quien es el responsable de la gestión del *backlog* del producto.
 - c) El *product owner* podrá ser solamente el cliente.
 - d) En proyectos que desarrollan productos genéricos para el mercado Scrum plantea que no hay *product owner*.
9. Usted forma parte del equipo de arquitectura de una organización gubernamental que está encargado del desarrollo y evolución de un gestor documental denominado Expediente Electrónico (EE). El EE es un software para gestionar los expedientes de diversos organismos del Estado en formato digital. Este sistema busca solucionar algunos de los problemas de interoperabilidad entre organismos, que surgen por tener cada uno su propia herramienta de seguimiento de expedientes. Este sistema a su vez debe integrarse con otros componentes independientes, como ser: (1) una solución para generar reportes, (2) otra para el modelado de procesos de negocio que hacen uso de los expedientes, y (3) un editor que permita diseñar el formato de los expedientes electrónicos. Actualmente su equipo está discutiendo el diseño arquitectónico de este sistema. Indique cuál de los siguientes patrones arquitectónicos resulta más adecuado al sistema planteado:
- a) Arquitectura entre pares (peer-to-peer)
 - b) Arquitectura maestro-esclavo
 - c) Arquitectura de repositorio
 - d) Arquitectura de tubería y filtros
10. Como líder del equipo de arquitectura en una agencia de transformación digital del Estado, se le presentan los siguientes desafíos en busca de la eficiencia tanto en procesos como en costos: definir pautas para mejorar el departamento de arquitectura de la agencia y evolucionar un sistema legado que tienen instalado localmente en algunos de los organismos estatales para gestionar sus propios trámites. Para dar solución a este último desafío se debería tener una solución que permita diseñar y gestionar los trámites para que puedan ser realizados 100% en línea y un sistema de agenda para ser utilizado por todos los organismos. Con ello en mente se puso a investigar la opción de contar con un sistema provisto bajo la modalidad de Software como servicio (*Software as a Service, SaaS*). En su investigación descubre que:
- I. Para pasar a SaaS es importante tener en cuenta los siguientes factores: escalabilidad, configuración y multitenencia (*multi-tenancy*).
 - II. El SaaS es una opción que puede ayudar a optimizar costos ya que se ahorraría en el mantenimiento y evolución de múltiples sistemas que hacen lo mismo (tienen el mismo propósito).
 - III. SaaS es una forma de desarrollar software como clientes gruesos que ejecutan de forma local, en el *desktop* de cada usuario con el fin de no sobrecargar el servidor.
 - IV. Para que el software sea escalable Somerville sugiere, como pauta general, que se diseñen bases de datos de forma tal que permitan un bloqueo completo de registros de la base de datos para evitar inconsistencias.

Seleccione la opción correcta:

- a) Solo las afirmaciones I y III son correctas.
- b) Solo las afirmaciones I y II son correctas.
- c) Solo las afirmaciones I, II y III son correctas.
- d) Solo las afirmaciones I, III y IV son correctas.

11. Usted fue contratado para una consultoría en una empresa de desarrollo de software porque en los últimos proyectos de desarrollo se produjeron diversos problemas cuya causa ha sido el abordaje del proyecto y la realización del diseño arquitectónico de forma aislada. Como parte de la consultoría se va a elaborar un plan de mejora que considere aspectos metodológicos (procesos/métodos) vinculados al diseño de software, así como definiciones generales de diseño que ayuden al momento de trabajar en el mismo. Los productos desarrollados son sistemas empresariales complejos que involucran a su vez varios sistemas distribuidos pero que en general son similares en cuanto a sus funcionalidades.

Teniendo en cuenta el contexto presentado, qué recomendaciones entiende que pueden ser más apropiadas para incluir en el plan:

- I. Establecer como pauta diseñar arquitecturas tomando como referencia la clasificación de Sommerville, seleccionando un único nivel de abstracción correspondiente a la arquitectura en pequeño (*Architecture in the small*), así el equipo puede hacer foco en los programas individuales para así mejorar la documentación y generar un mejor diseño que pueda ser reutilizable.
- II. Resulta razonable tener en cuenta fundamentos de la ingeniería de software basada en componentes (CBSE), como ser: los componentes independientes son completamente especificados por sus interfaces, contar con un *Middleware* que de soporte a la integración de los componentes y estándares que faciliten la integración.
- III. Incorporar en el proceso de desarrollo actividades específicas de evaluación y validación del diseño arquitectónico propuesto, así como también tener en cuenta otras arquitecturas de otros proyectos que podrían adecuarse/adaptarse al nuevo sistema.
- IV. Para la documentación de todos los sistemas usar la vista 4 + 1, donde algunas de sus vistas son la vista de procesos -para visualizar la interacción de procesos en tiempo de ejecución-, y la vista de desarrollo -para visualizar las abstracciones clave en el sistema como objetos o clases de objetos-.

Seleccione la opción correcta:

- a) Solo las afirmaciones II y III son correctas.
- b) Solo las afirmaciones I, II y III son correctas.
- c) Solo las afirmaciones II y IV son correctas.
- d) Solo las afirmaciones I, III y IV son correctas.

12. Usted pertenece a un equipo de SCM de una empresa que brinda dos tipos de productos de software con un amplio alcance, uno de ellos es provisto de forma centralizada y masiva, y el otro es un software para ser instalado localmente en cada uno de los clientes. El primero, es un software de búsqueda semántica (similar a cómo funciona Google) que tiene tres componentes: uno que transforma documentos pdf a texto, otro que es el buscador en sí mismo y un tercer componente que se encarga de generar el modelo (que se usa para las búsquedas) y re-entrenarlo. El segundo software es para la Gestión de recursos humanos en las empresas. Este último se desarrolla en Suecia, Uruguay e India.

Teniendo en cuenta esta realidad, indique las afirmaciones más apropiadas:

- I. Es necesario definir un Comité de control de cambios (Change Control Board - CCB) para el software de Búsqueda semántica compuesto por todos los clientes del sistema para que entre todos decidan qué conviene hacer con cada cambio.
- II. Para el sistema de gestión de RRHH se debe contar con el registro de las versiones liberadas (*releases*) distribuidas a cada cliente para poder brindar un soporte adecuado.
- III. En ambos casos, cada "*release*" de software debe incluir únicamente la versión ejecutable del software y los archivos de configuración.
- IV. En general, para software del mismo tipo del de "Búsqueda semántica" (que apuntan a un público masivo), el proceso de preparar y distribuir software es un proceso que incluye actividades como la generación de material publicitario.
- V. Para poder realizar una integración frecuente de software es necesario que todos los componentes estén identificados con el mismo número de versión.

Seleccione la opción correcta:

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

13. Supongamos que usted quiere probar la siguiente pieza de código.

```

If (temperature < MAX-T OR (presion < MAX-P AND humedad < MAX-H))
    code_err = OK;
Else
    If (presion >= MAX-P) Code_err = "Presion-muy-alta"
    Else
        Code_err = "Humedad-muy-alta"
Return(Code_err)
}

```

¿Cuántos casos de prueba son necesarios (la mínima cantidad) para asegurar un cubrimiento de decisión y cuántos caminos simples tiene el grafo de flujo de control?

- a) 2 caminos - 3 casos de prueba
- b) 3 caminos - 2 casos de prueba
- c) 3 caminos - 3 casos de prueba
- d) 3 caminos - 6 casos de prueba

14. Sobre las pruebas de regresión. Considere las siguientes afirmaciones:

- I. Se centran en detectar defectos introducidos al realizar modificaciones al código.
- II. En un proceso iterativo e incremental, es razonable ejecutar pruebas de regresión al final de cada iteración para intentar asegurar que el software no se haya degradado.
- III. Son muy útiles durante la evolución y mantenimiento del software para asegurar que no se hayan introducido defectos en partes del software ya probadas y que funcionan correctamente en producción.
- IV. Son recién necesarias a partir del momento en que el software está en mantenimiento.

Seleccione la opción correcta:

- a) Solo las afirmaciones II y IV son correctas.
- b) Solo las afirmaciones I, II y III son correctas.
- c) Solo las afirmaciones I, III y IV son correctas.
- d) Todas las afirmaciones son correctas.

15. Sobre la gestión de las pruebas, seleccione la afirmación INCORRECTA:

- a) Contar con un equipo de testing independiente del de desarrollo tiene varias ventajas. Entre ellas: evitar los problemas relacionados a la “psicología de las pruebas” (donde el autor del programa comete los mismos errores al probarlo que al codificarlo), contar con habilidades específicas para cierto tipo de pruebas (por ejemplo, seguridad) de las cuales pueden carecer los desarrolladores.
- b) Contar con un equipo independiente de testing puede traer problemas. En particular, los desarrolladores podrían perder la responsabilidad por la calidad del software entregado y esperar que los errores y oportunidades de mejora los encuentre el equipo de testing.
- c) Es importante que los casos de prueba se prioricen. Para priorizar los casos de pruebas se pueden utilizar distintos criterios tales como: priorizar en base al uso de las funciones, priorizar en base a los riesgos de que ocurra una falla o en base a la prioridad de los requisitos.
- d) La planificación de las pruebas debe realizarse de forma independiente a la planificación del equipo de desarrollo. Esto es necesario para que el equipo de testing no sea un cuello de botella al momento de querer liberar una versión o incremento del producto.

16. Cuando se sustituye un sistema viejo por uno nuevo:

- a) La estrategia de big-bang permite gestionar mejor los riesgos inherentes a la sustitución.
- b) La migración de datos (si hubiera) no puede realizarse si la sustitución se realiza con un procesamiento en paralelo.
- c) Una estrategia paulatina implica que el sistema viejo comience a operar como “secundario” y como control ante un fallo del sistema nuevo.
- d) Ninguna opción es correcta.

17. Cuando un software está en producción, pueden haber pedidos de cambio que impliquen un trato urgente. Estos pedidos de cambio pueden surgir debido a:

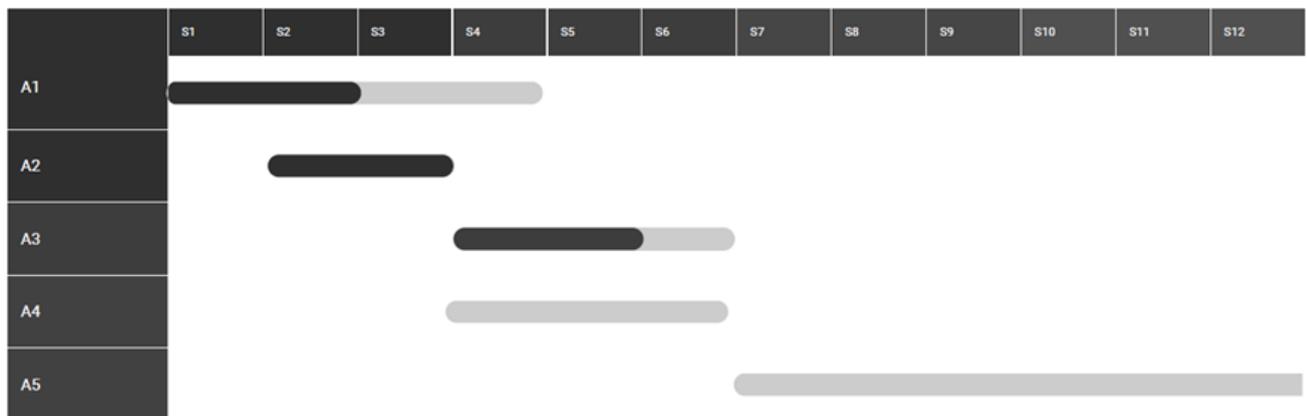
- a) Si una falla seria es detectada y debe ser reparada para que el software pueda funcionar normalmente o para abordar una vulnerabilidad de seguridad seria.
- b) Si los cambios en el entorno donde el software opera tienen efectos inesperados, los cuales interrumpen el funcionamiento normal.
- c) Si se producen cambios imprevistos en el negocio donde el software da soporte, como ser la aparición de nuevos competidores o la introducción de una nueva legislación que afecte al software.
- d) Todas las respuestas anteriores son correctas.

18. Respecto a la comunicación en un proyecto:

- Al comunicarnos, no es suficiente decir lo que queremos comunicar con las palabras, el tono con el que lo decimos, así como el lenguaje corporal deben ser consistentes.
- Algunas barreras de comunicación son: personalidad, intereses, suposiciones sobre el receptor/emisor, hacer contacto visual y escucha selectiva.
- En comunicación, lo que mayor impacto tiene en el mensaje son las palabras, por esto es importante pensar muy bien lo que vamos a decir y no tanto cómo lo decimos.
- La comunicación puede ser un problema debido a la cantidad de integrantes distribuidos y los canales de comunicación se calculan como $n*n/2$, donde n es la cantidad de integrantes del equipo.

19. Sea el siguiente cronograma de 12 semanas donde se refleja la duración de las tareas (en gris) y el avance actual (en negro), en el que las tareas tienen el siguiente costo total planificado:

A1 - 160
 A2 - 80
 A3 - 120
 A4 - 120
 A5 - 240



Si el valor planificado para el fin de la semana 4 es 300, y asumiendo que estamos planificando con la misma técnica de medición del avance para todas las tareas, ¿cuál es esta técnica?

- Porcentaje de avance.
- Fórmula fija 25/75.
- Fórmula fija 0/100.
- Hitos con peso: un hito de comienzo que asigna 10 % del valor, otro por la mitad que asigna 50 % del valor y otro al finalizar que asigna el 40 % restante.

20. En relación con la etapa de identificación de riesgos:

- Es el proceso de determinar los riesgos que pueden afectar al proyecto y documentar sus características.
- (a) y este proceso no tiene por qué ser iterativo si al inicio del proyecto se identifican todos los riesgos de mayor impacto.
- En esta etapa solo pueden participar stakeholders del proyecto
- Para cada riesgo identificado se debe evaluar la probabilidad de ocurrencia y el impacto.

21. Respecto a las estimaciones de duración o esfuerzo:

- a) La incertidumbre al estimar la duración de un proyecto varía dependiendo de la etapa en la que este se encuentre.
- b) En la fase inicial de los proyectos, las estimaciones por lo general tienen un alto margen de incertidumbre. Sin embargo, en la medida que el proyecto avanza, dicha incertidumbre irá disminuyendo, incluso en proyectos con requisitos inestables y tecnología desconocida.
- c) Aun sabiendo de cuántos desarrolladores se dispone y su dedicación, la productividad personal de los desarrolladores varía tanto que no es posible estimar la duración de un proyecto si no se sabe quién va a trabajar en él.
- d) Las estimaciones son más certeras si se realizan en un proyecto bajo metodología ágil.