

Instrucciones

- Cada pregunta múltiple opción respondida correctamente tiene un valor de 2,5 puntos.
- Cada pregunta múltiple opción respondida incorrectamente resta 0,83 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 una hora y cuarenta y cinco minutos.
- El puntaje total del parcial es de 45 puntos.

1. Seleccione la afirmación correcta:

- a) El incremento, en los últimos años, de la complejidad de los sistemas de software es uno de los tantos factores que están vinculados a la calidad del software que se produce.
- b) (a) y la ingeniería de software, como disciplina, no considera temas éticos, ya que esto es una rama de la ingeniería en general y también de estudio actual en filosofía.
- c) (a) y, al desarrollar software, fallar en el uso de las técnicas y métodos de la ingeniería de software no ocasiona problemas ni de calidad ni de cronograma si se usan métodos ágiles. Esto es debido a que los métodos ágiles son métodos que normalmente usan retrospectivas y validaciones, lo que permite corregir rápidamente los problemas.
- d) Los procesos de desarrollo de software basados en incrementos aumentan la complejidad de la gestión de la configuración respecto a procesos basados en iteraciones.

2. Seleccione la afirmación correcta:

- a) Los procesos de desarrollo de software actuales (o modernos) no definen roles. Esto se debe, como se ha visto de forma empírica, a que los desarrolladores deben poder autogestionarse para avanzar ágilmente.
- b) Algunos procesos de desarrollo de software se componen de ciclos de refinamiento continuo (por ejemplo, el desarrollo en cascada), sobre todo en las fases de requisitos de software y transferencia al cliente.
- c) Scrum propone revisar en cada *sprint* la forma en la cual el equipo de desarrollo trabajó durante el *sprint* y reflexionar sobre cómo se podría haber trabajado mejor.
- d) Los procesos ágiles se centran en la arquitectura de software, de forma de que en cada iteración la arquitectura sea definida al principio y luego refinada.

3. Una empresa es contratada para desarrollar un cierto producto de software. Luego de trabajar en recolectar al menos una idea general de los requisitos, los líderes del proyecto de desarrollo de software estiman el tamaño del proyecto y el esfuerzo. Dado el tamaño del equipo de desarrollo, estiman que el proyecto para la construcción del producto de software pedido llevará alrededor de un año y medio de trabajo.

Seleccione la afirmación correcta:

- a) Dado que el trabajo va a durar un año y medio, es razonable usar un proceso tradicional de desarrollo de software en lugar de uno ágil, ya que los planes serán de mucho valor.
- b) Dado que el trabajo va a durar un año y medio, es razonable utilizar un método ágil, ya que al durar ese tiempo va a haber muchos cambios en los requisitos durante el desarrollo.
- c) Los datos sobre el proyecto que nos da esta pregunta (que se estimó el tamaño y esfuerzo, y que dado el tamaño del equipo de desarrollo va a llevar un año y medio la construcción) son insuficientes para poder indicar qué proceso de desarrollo utilizar.
- d) Con esta realidad planteada, el siguiente paso es desarrollar una arquitectura en capas buscando flexibilidad, debido a la duración prolongada del proyecto (aproximadamente un año y medio).

4. Seleccione la afirmación correcta:
- a) Los procesos iterativos e incrementales logran liberar incrementos (versiones) al cliente al menos una vez por mes.
 - b) Las iteraciones breves (por ejemplo, *sprints* de dos semanas) y la planificación a muy corto plazo para el desarrollo no siempre encajan (funcionan bien) con los planes más a largo plazo del negocio o de marketing.
 - c) En la programación por pares, dos desarrolladores hacen en paralelo la misma porción de software al mismo tiempo cada uno en su computadora. Al finalizar, se corren pruebas para conocer cuál porción de software (de las dos desarrolladas) se comporta mejor, se elige una y se analizan posibles mejoras.
 - d) Los métodos ágiles se basan fuertemente en tener un cliente o usuario en el mismo lugar físico que el equipo de desarrollo. XP llamó a esta práctica *cliente en el lugar*. El cliente o usuario está siempre disponible en el horario del equipo de desarrollo de software.
5. Seleccione la afirmación correcta:
- a) Un proceso de desarrollo de software presenta las actividades que debe seguir el equipo de desarrollo de software para realizar el desarrollo de cierto proyecto particular lo mejor que se puede según el conocimiento actual de la empresa de desarrollo. Por eso, debe seguirse el proceso de forma disciplinada y no realizarle cambios durante el transcurso del proyecto.
 - b) El modelo CMMI —por sus características y, a su vez, también debido a los métodos ágiles y al tiempo que pasó desde la última versión de este modelo— no puede ser utilizado para la mejora de procesos de desarrollo de software actuales o modernos.
 - c) Los prototipos de software sirven mucho para la etapa de puesta en producción (liberación) del software. Esto es porque con ellos pueden hacerse demostraciones a los usuarios de cómo deben usar el sistema que se está instalando. En etapas tempranas los prototipos no son de utilidad, ya que los usuarios tienden a pensar que el producto ya está pronto y debería ser disponibilizado, lo que causa discusiones y riesgos de relacionamiento cliente-usuario-empresa de desarrollo que serán difícil de resolver.
 - d) El desarrollo incremental busca, entre otras cosas, construir software mediante incrementos. Cada incremento sirve o para brindar funcionalidad al cliente (en el sentido de liberar, poner en producción el incremento) o para tener retroalimentación del cliente y de los usuarios, incluso si el incremento no se pone en producción.
6. Sobre la construcción de software, seleccione la afirmación correcta:
- a) La planificación de la construcción incluye, entre otras cosas, cómo van a ser construidos e integrados los componentes de software.
 - b) La construcción es una etapa del desarrollo de software que normalmente no se realiza en la práctica profesional (es decir, en la industria de software), ya que se pasa directo a la programación luego del diseño.
 - c) (b) y la construcción efectiva y eficiente que se ha logrado mostrar a través de los últimos años y que sigue vigente es mediante el uso del lenguaje de programación Java.
 - d) (a) y muchas veces, uno de los efectos del mantenimiento y evolución del software es el deterioro del diseño y del código.

7. Sobre la construcción de software, según el SWEBOOK, seleccione la afirmación correcta:
- El reuso es una actividad que comprende utilizar (reutilizar) software ya existente como parte del desarrollo de software que se está realizando.
 - (a) y comprende también el construir software que pueda ser luego reutilizado.
 - El reuso no está vinculado al área de conocimiento de construcción; el reuso es razonable durante el diseño y por eso está vinculado al área de conocimiento de diseño.
 - La construcción de software, en procesos de desarrollo de software en cascada, se realiza sin revisiones o inspecciones de código.
8. Se tiene un sistema que se quiere actualizar a una nueva tecnología. No se cuenta con documentación del sistema. El sistema actual tiene una interfaz de usuario. Se dispone de poco tiempo. ¿Qué técnica de obtención de los requisitos sería la más adecuada para obtener la funcionalidad a implementar?
- El análisis de la interfaz de usuario del sistema viejo para obtener requisitos de allí.
 - Una encuesta a todos los posibles usuarios del nuevo software.
 - Entrevistas con los usuarios del sistema anterior.
 - Una tormenta de ideas con los usuarios del sistema anterior.
9. En cuanto al grado de corrección y completitud de los modelos utilizados para analizar los requisitos:
- Si se usan como medio de facilitar la discusión acerca de sistemas existentes o propuestos, deberán ser completos y correctos.
 - Si se usan como una manera de documentar un sistema existente, deben ser una representación completa del sistema.
 - Si se usan como una descripción del sistema para generar una implementación, la descripción tiene que ser correcta, aunque no es necesario que sea completa.
 - Depende del propósito para el cual se use el modelo.
10. Indique cuáles de las siguientes afirmaciones pueden considerarse requisitos no funcionales para un software a desarrollar:
- El software deberá comunicarse con el sistema de la DNIC.
 - El software deberá correr en tablets y en la web.
 - El software deberá ser construido utilizando RubyOnRails.
 - La funcionalidad de Buscar un Activo no deberá demorar más de 3 segundos para mostrar los resultados.
 - El software deberá proveer la gestión de usuarios y permisos.

Seleccione la opción correcta:

- Solo IV es correcta.
- Solo II, III y V son correctas.
- Solo I, II, III y IV son correctas.
- Todas son correctas.

11. Las historias de usuario

- a) Son descripciones cortas y de alto nivel de las funcionalidades expresadas en los términos del cliente, por lo que no hay forma de agregarle detalles útiles para los implementadores o verificadores.
- b) Se utilizan solo en desarrollo ágil.
- c) Consisten en una descripción breve de un requisito, que es utilizada como base para la implementación y para planificar.
- d) Según la definición de eXtremme Programing, una historia de usuario describe una funcionalidad que, por sí misma, aporta valor al usuario o al equipo de desarrollo.

12. La validación de requisitos es el proceso por el cual

- a) se busca determinar si los requisitos relevados son consistentes con las necesidades del cliente.
- b) se busca determinar si los requisitos están escritos sin ambigüedades ni inconsistencias y pueden ser implementados con el presupuesto y la tecnología disponible.
- c) se busca determinar si el software construido pasa las pruebas de aceptación del sistema.
- d) el equipo de verificación busca determinar si los requisitos especificados están correctamente escritos, mediante la generación de casos de prueba.

13. Usted es parte del equipo de arquitectura de soluciones en una empresa de desarrollo de software. Le vinieron a proponer como proyecto el desarrollo de un sistema para una tienda inteligente donde los clientes se autogestionan la compra. Es importante utilizar la tecnología para mejorar la experiencia de compra. La clave para lograrlo es contar con una importante inversión en infraestructura y un sistema robusto. Como recomendación le sugieren que se ponga a investigar acerca de sistemas distribuidos. En su investigación descubre que:

- I. según Coulouris, dos de los principales beneficios de desarrollar sistemas distribuidos son la ejecución concurrente y la posibilidad de compartir recursos.
- II. el patrón maestro-esclavo es una buena opción para garantizar tiempos de respuesta de interacción en sistemas de tiempo real.
- III. puede hacer uso del patrón cliente-servidor multinivel para implementar las diferentes capas de un sistema y así distribuir el procesamiento.
- IV. en el patrón de componentes distribuidos los únicos tipos de componentes permitidos son los middlewares.
- V. los sistemas distribuidos, por su naturaleza, son más complejos de probar que los centralizados.
- VI. el patrón entre pares (peer-to-peer) es el más adecuado para ser usado cuando se quiere que los clientes hagan sus compras en el futuro sistema de la tienda de forma eficiente.

Seleccione la opción correcta:

- a) I, III, IV, VI: son correctas
- b) I, II, III, V: son correctas
- c) I, III, V, VI: son correctas
- d) Todas son correctas

14. Como responsable de la arquitectura de un sistema, algunos aspectos que debo tener en cuenta al momento de abordar el proceso de diseño y el propio diseño son:

- I. el proceso de desarrollo de software que se está aplicando, para prever los momentos en los cuales es necesario que se trabaje sobre la arquitectura.
- II. identificar los principales componentes del sistema y sus relaciones para entender cómo se puede organizar el sistema.
- III. aplicar el principio de abstracción para que los componentes se puedan implementar separadamente, procurando que el cambio en uno tenga impactos mínimos en el resto.
- IV. que es mejor comenzar el diseño arquitectónico en etapas avanzadas del proceso de desarrollo porque su cambio requiere mucho esfuerzo y es costoso y no es necesario tenerlo pronto en etapas tempranas.
- V. analizar el costo/beneficio en el reuso de componentes existentes para determinar si conviene integrarlos como parte del desarrollo de software.

Seleccione la opción correcta:

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

15. Un sistema se va a desarrollar aplicando un enfoque ágil. Durante el proyecto se requiere la puesta periódica en producción cada 2 iteraciones. El proyecto está planificado con una duración total de 6 iteraciones.

Seleccione la afirmación correcta:

- a) Al estar aplicando un proceso ágil, no es aceptado que en etapas tempranas el proceso haga foco en definir una arquitectura general del sistema.
- b) Teniendo en cuenta los cambios que puedan surgir al relevar los requisitos durante las distintas iteraciones, en este caso, se sugiere utilizar una arquitectura de componentes distribuidos debido a los beneficios que proporcionan respecto a la apertura y escalabilidad.
- c) La arquitectura del sistema no podrá estar totalmente definida hasta luego de la última iteración debido al relevamiento continuo de requisitos que hacen surgir cambios en la arquitectura antes de cada puesta en producción.
- d) La arquitectura del sistema deberá soportar los requisitos funcionales y no funcionales, sería bueno definirla en etapas tempranas del proceso de desarrollo de software y puede ir evolucionando a medida que se va avanzando en el proyecto.

16. Se quiere elaborar un servicio nacional que brinde, de forma centralizada, el domicilio de las personas, con una seguridad multinivel para aislar riesgos vinculados a la protección de datos personales. Se conoce que siempre se va a requerir la misma forma de ver e interactuar con los datos y que la presentación no va a cambiar. También, que debe existir una clara separación y encapsulamiento de responsabilidades con roles definidos dentro del sistema.

Indique cuál de los siguientes patrones arquitectónicos se ajusta mejor a la realidad presentada:

- a) Arquitectura de tubería y filtro
- b) Arquitectura en capas
- c) Arquitectura de repositorio
- d) Arquitectura en capas donde la capa lógica tiene una arquitectura de tubería y filtros.

17. Si utilizo el modelo 4+1 para representar distintas vistas de la arquitectura, cuál de las siguientes afirmaciones es más completa:

- a) Puedo utilizar la vista física para mostrar cómo los componentes del software se distribuyen en los equipos (hardware) involucrados, por ejemplo, cuando aplico un patrón arquitectónico cliente-servidor en un sistema distribuido
- b) La vista de casos de uso / escenarios sirve para conocer las abstracciones clave en el sistema como objetos o clases de objetos
- c) a) y la vista de proceso, para visualizar la interacción de procesos en tiempo de ejecución.
- d) b) y la vista de desarrollo, para visualizar cómo el software se descompone para el desarrollo.

18. Dado el contexto planteado en la letra del Laboratorio 1, seleccione la opción correcta:

- a) Como el sistema a desarrollar se debe integrar con los sistemas de DNIC y SINAE, el modelo de proceso de integración y configuración parecería ser el más adecuado a utilizar en el proceso de desarrollo.
- b) La técnica de casos de uso sería una técnica efectiva de relevamiento de requisitos de la aplicación «Gestión de solicitudes, quejas y reclamos técnicos», pero no de especificación de requisitos.
- c) Pensar en mecanismos para involucrar a los ciudadanos (o una muestra representativa de estos) en el relevamiento y validación de los requisitos de la aplicación móvil no sería algo deseable, ya que lo importante es que el cliente (GobDep) sea quien defina y valide los requisitos a implementar.
- d) Si se tuviera el diagrama de CU completo de todo el sistema, el sistema de finanzas, el sistema de control de stock y el sistema de gestión de RR. HH. del GobDep deberían ser modelados como actores externos al sistema.