

Tests Fundamentos de Ingeniería del Software

1.- Una mala solución para remediar el retraso en la entrega de un proyecto software es la llamada "horda mongoliana". Verdadero

La horda mongoliana consiste en contratar programadores extra cuando se acerca una fecha de entrega y el programa no está listo, pensando que se acelerará el proceso. Sin embargo, lo que ocurre es que hay que explicar el código y su funcionamiento así como el proyecto a los nuevos programadores lo que se traduce en tiempo.

2.- Una asociación es una conexión significativa y relevante entre conceptos. Verdadero

Es justamente la definición de asociación que se utiliza para un modelo conceptual.

3.- Un nivel de acoplamiento alto y de cohesión bajo en un módulo garantiza un diseño de calidad. Falso

Garantiza independencia funcional. Para que se garantice un diseño de calidad se tienen que cumplir los principios de abstracción, división de problemas y modularidad, ocultamiento de información y independencia funcional.

4.- Un modelo de casos de uso se centra en las necesidades que el usuario espera lograr al utilizar el sistema. Verdadero

5.- Un modelo de casos de uso lo componen los diagramas de casos de uso y la especificación de actores y casos de uso. Verdadero

Lo componen los actores, los casos de uso y las relaciones (actor-actor, actor-caso de uso, caso de uso-caso de uso). Para la representación y descripción de estos elementos se usan diagramas de caso de uso de UML y plantillas estructuradas para los actores y casos de uso.

6.- Un concepto no debe incluir los atributos de otros conceptos que indiquen las relaciones entre ellos. Verdadero

Relacionar conceptos con asociaciones y no con atributos.

7.- Un caso de uso puede ser iniciado por un actor o por un usuario. Falso

Los casos de uso son iniciados por un actor que, junto con otros actores, intercambian datos o control con el sistema a través de él.

8.- Un caso de uso produce algo de valor para un actor. Verdadero

Un caso de uso tiene que tener utilidad concreta para algún actor y describir un fragmento de funcionalidad del sistema de principio a fin, y tienen que acabar y proporcionar un resultado.

9.- Un caso de uso esencial describe que hace el sistema como respuesta a una petición de algún actor, pero no como lo hace. Verdadero

Un caso de uso esencial está expresado de forma abstracta, contiene poca tecnología y pocos detalles de diseño.

10.- Si una función del sistema no cambia nada de lo especificado en el modelo conceptual su contrato no tendrá poscondiciones. Verdadero

Si una función del sistema no cambia nada de lo especificado en el modelo conceptual, no modifica el estado del sistema ni ninguno de los objetos del modelo conceptual, luego no hay poscondiciones.

11.- Para incorporar generalizaciones es necesario encontrar clases conceptuales con elementos comunes. Verdadera

Como habla de clases conceptuales suponemos que hablamos del modelo conceptual, y para agregar una generalización en el modelo conceptual se deben encontrar clases conceptuales con elementos comunes, definir las relaciones de superclase y subclase y verificar que la subclase cumple la regla del 100% y la regla del es-un.

12.- Para elaborar el modelo de análisis es fundamental el modelo de casos de uso. Verdadero

El modelo de análisis parte del modelo de casos de uso, la lista de requisitos y el glosario de términos.

13.- Los tipos de requisitos son funcionales, no funcionales y FURPS+. Falso

Entre otras cosas no se tiene en cuenta los requisitos de información.

14.- Los requisitos no funcionales suponen limitaciones para el diseño de un sistema software. Verdadero

Los requisitos no funcionales especifican aspectos técnicos que debe incluir el sistema, bien sean restricciones o calidades, lo que implica limitaciones (También se pueden ver como restricciones de los requisitos funcionales).

15.- Los requisitos no funcionales determinan los objetivos del diseño. Falso

Al identificar los objetivos del diseño, se identifican las cualidades deseables del sistema, se obtienen a partir de los requisitos no funcionales (no tienen porque coincidir exactamente), se selecciona un pequeño conjunto de objetivos de diseño que el sistema debe satisfacer necesariamente y se adquieren compromisos, puesto que muchos objetivos de diseños son contrapuestos.

16.- Los prototipos siempre se transforman hasta convertirse en el programa que se entrega al cliente. Falso

No siempre se entrega la versión final de un prototipo. A veces un prototipo simplemente se genera para probar una funcionalidad en concreto y después se desecha.

17.- Los modelos de AER son: modelo conceptual, diagramas de casos de uso y los contratos de las operaciones principales. Falso

Son los diagramas de conceptos (modelos conceptuales), los diagramas de secuencia del sistema y los contratos.

18.- Los diagramas de interacción y los diagramas de actividad UML son herramientas de diseño que permiten representar lo mismo, son equivalentes. Falso

El diagrama de actividad sirve para la descripción del comportamiento que tiene un conjunto de tareas o procesos, mientras que el diagrama de interacción utiliza los patrones de diseño para describir los principios fundamentales del diseño de objetos y la asignación de responsabilidades.

19.- Los diagramas de actividad de UML es una herramienta muy adecuada para el diseño del flujo de control. Verdadero

Los diagramas de actividad se pueden usar para representar los flujos de un caso de uso de forma gráfica.

20.- Los actores tienen que ser necesariamente los identificados como usuarios del Sistema. Falso

Pueden ser también usuarios del producto.

21.- Los actores representan roles que son interpretados por personas, dispositivos, otros sistemas... cuando el sistema está en uso. Verdadero

Los actores especifican roles que adoptan entidades externas cuando interactúan con el sistema, y pueden ser personas, dispositivos E/S, sistemas informáticos externos y temporizadores o relojes.

22.- Los siguiente es una poscondición correcta "se creó una lista en la que se incluye el nombre del cliente, dirección y teléfono. Que se proporciona como salida de la operación". Falso

Las poscondiciones no hablan de que incluye un objeto de se crea ni que tipo de objeto es, eso se hace en la salida. Las poscondiciones nos dice si se ha creado un enlace entre dos objetos o si se ha creado un objeto a que clase pertenece.

23.- Lo siguiente es un requisito NO funcional de facilidad de uso "el entorno debe avisar al usuario mediante email tras días antes de que finalice el plazo del préstamo". F

Especifica una función que un sistema, o componente de un sistema, debe ser capaz de llevar a cabo, es decir, es un requisito funcional.

24.- Lo siguiente es un recurso funcional "las reservas de préstamos de libros caducan a los 10 días a partir del momento que el libro esté a disposición del usuario". Verdadero

Misma justificación que el anterior.

25.- Las vías de comunicación o enlaces entre objetos en un diagrama de colaboración son bidireccionales. Verdadero

El diagrama de colaboración se centran más en los elementos dentro de un sistema que en el flujo de los mensajes.

26.- Las tareas principales de la ingeniería de requisitos son detección, análisis, especificación, revisión y reacción de requisitos. Falsa

Las actividades de la ingeniería de requisitos son análisis, obtención, especificación y validación (la validación genera el documento de requisitos del sistema.)

27.- Las relaciones que se dan entre casos de uso son la dependencia y la generalización. Falso

Son la dependencia, la generalización y la extensión.

28.- Las relaciones entre los casos de uso pueden ser asociación, generalización y dependencia. Falso

La asociación no es una relación entre casos de uso, es una relación entre caso de uso y actor.

28.- Las relaciones entre actores y casos de uso son la asociación y la dependencia. Falso

La dependencia no es una relación entre caso de uso y actor.

29.- La primera tarea del diseño es encontrar el diseño de la arquitectura del sistema. Verdadero

Las tareas son, en orden, diseñar la arquitectura, diseñar los subsistemas(diseñar casos de uso y diseñar clases) y revisión.

30.- La navegabilidad de las asociaciones en el diagrama de clases del diseño se obtiene teniendo en cuenta la dirección en los envíos de mensaje en los diagramas de comunicación. Verdadero

Es justo como se obtiene la navegabilidad en el diagrama de clases del diseño.

31.- La forma más directa de identificar casos de uso es identificando los objetivos y necesidades de los actores del sistema. Verdadero

32.- La arquitectura cliente-servidor favorece la escalabilidad de los sistemas software, porque permite la reconfiguración añadiendo clientes y servidores extra. Verdaderos

Entre las ventajas de una arquitectura cliente servidor se encuentran, que se pueden añadir fácilmente servidores o clientes extras.

33.- Es posible que en un caso de uso no tenga que intervenir el sistema software a modelar. Falso

No tiene sentido, pues los casos de uso representan escenarios que se pueden dar al interactuar un actor con el sistema, luego en todos ellos participa el sistema.

34.- En un diagrama de secuencia del sistema pueden aparecer tantos objetos como necesitemos para modelar la interacción entre ellos y con los actores. Falso

En un diagrama de secuencia no hay objetos (creo).

35.- En los diagramas de clases de diseño pueden aparecer relaciones de dependencia. Verdadero

36.- En los diagramas de clases de diseño no se deben representar las relaciones de dependencia entre clases, solo se deben representar las de asociación y de generalización. Falso

Las relaciones de dependencia también hay que representarlas.

37.- En el modelo conceptual hay que definir los atributos y los métodos de todas las clases. Falso

En un modelo conceptual no hay métodos, los métodos se ponen en el diagrama de clases del diseño.

38.- En el diagrama de clases del diseño pueden aparecer nuevas clases, es decir, que no estén en el modelo conceptual. Verdadero

Pueden haber más clases si así se considera necesario.

39.- El resultado del diseño de la arquitectura del software es un conjunto de subsistemas y las relaciones entre ellos. Verdadero

Las actividades del diseño arquitectónico son identificar los objetivos del diseño, determinar la arquitectura software, y los correspondientes subsistema, modelar la arquitectura software y refinar la descomposición en subsistemas.

40.- El proceso unificado es un modelo de proceso dirigido por casos de uso. ¿Verdadero?

41.- El número de operaciones principales de un sistema es el mismo que el número de casos de usos que tengamos. Falso

Las operaciones principales del sistema se identifican y normban a partir de las descripciones de los casos de uso, pero no tienen porque coincidir (por ejemplo, un mismo caso de uso puede contener varias operaciones).

42.- El número de módulos de un sistema software debe ser cuantos más mejor, pues así garantizamos la independencia modular de cada uno de ellos. Falso

Debe ser el justo y necesario, más no siempre es mejor.

43.- El modelo de casos de uso se usa exclusivamente para la obtención de requisitos. Falso

Se usa también en el análisis y especificación de requisitos. Además en la fase de obtención de requisitos es donde se genera el modelo de casos de uso.

44.- El modelo de casos de uso permite determinar con facilidad los requisitos no funcionales del sistema. Falso

La lista de requisitos y el modelo de casos de usos se obtienen en la misma etapa (obtención de requisitos) y es un proceso difícil.