Software Testing Fundamentals

Reactivos

Pregunta 1

¿Cuál es el principal propósito del software testing?

Opciones:

a. Minimizar el riesgo de mal funcionamiento del software.
b. Encontrar todos los errores posibles del software.
c. Garantizar que el software es 100% libre de errores.
d. Hacer que el software sea más rápido y eficiente.

Pregunta 2

¿Qué indica el principio de la paradoja del pesticida en el software testing?

Opciones:

a. Las pruebas deben realizarse solo al final del desarrollo.
b. Utilizar las mismas pruebas repetidamente no descubrirá nuevos errores.
c. Las pruebas de software no son necesarias para proyectos pequeños.
d. Los errores en el software son inevitables y no pueden ser resueltos.

Pregunta 3

¿Cuál de estos es un tipo de prueba de software?

Opciones:

a. Prueba de hardware.
b. Prueba de usuario.
c. Prueba de compatibilidad.
d. Prueba de diseño gráfico.

Pregunta 4

¿Cómo debería un tester abordar un error encontrado en una aplicación que no se reproduce consistentemente?

Opciones:

a. Cambiar el enfoque a otros errores más consistentes.
b. Informar el error solo si se reproduce en más del 50% de los intentos.
c. Ignorar el error ya que no es constante.
d. Documentar el error y los pasos exactos intentados, aunque sea inconsistente.

Pregunta 5

En una situación donde un software no cumple con un requisito no funcional de rendimiento, ¿cuál sería el siguiente paso adecuado?

Opciones:

a. Informar a los stakeholders que el requisito es irrealizable.
b. Cambiar el requisito para que coincida con el rendimiento actual.
c. Optimizar el código y los recursos para cumplir con el requisito.
d. Ignorar el requisito, asumiendo que los usuarios no notarán.

Pregunta 6

¿Cuál es el propósito de las pruebas de integración en el desarrollo de software?

Opciones:

a. Probar la funcionalidad individual de los componentes.
b. Evaluar cómo interactúan entre sí diferentes módulos o sistemas.
c. Determinar la eficiencia del código.
d. Verificar la estabilidad del sistema bajo condiciones extremas.

Pregunta 7

¿Qué caracteriza a las pruebas de aceptación del usuario (UAT)?

Opciones:

a. Confirmar que el software cumple con las expectativas y necesidades del usuario.
b. Verificar el cumplimiento de las especificaciones técnicas.
c. Evaluar el rendimiento del software en condiciones de carga.
d. Inspeccionar la calidad del código fuente.

Pregunta 8

En el software testing, ¿qué implica una prueba de regresión?

Opciones:

a. Verificar que los cambios recientes no afecten negativamente las funcionalidades existentes.
b. Evaluar la eficiencia del código después de una optimización.
c. Realizar pruebas exhaustivas en cada componente del software.
d. Probar el software en su totalidad en un entorno de producción.

Pregunta 9

En un proyecto de software, ¿cómo se debería actuar si se detecta que una prueba unitaria falla repetidamente?

Opciones:

a. Eliminar la prueba unitaria.
b. Ignorar el fallo, ya que puede ser un problema menor.
c. Revisar y corregir el código relacionado con la prueba fallida.
d. Continuar con las pruebas de integración sin resolver el fallo.

Pregunta 10

Si durante las pruebas de integración se encuentra que dos módulos no interactúan correctamente, ¿cuál es la mejor acción a seguir?

Opciones:

a. Reemplazar uno de los módulos por completo.
b. Ignorar el problema y continuar con las pruebas de sistema.
c. Cambiar los requisitos para que los módulos no necesiten interactuar.
d. Probar los módulos individualmente para identificar la fuente del problema.

Pregunta 11

¿Qué representa un "script de prueba" en comparación con un caso de prueba?

Opciones:

a. Un conjunto de requerimientos de software.
b. Una lista de errores conocidos en el software.
c. Un documento detallado de los pasos de prueba.
d. Un programa que ejecuta automáticamente las pruebas.

Pregunta 12

¿Qué factor no es típicamente evaluado en un caso de prueba?

Opciones:

a. Funcionalidad del software.
b. Estética de la interfaz de usuario.
c. Seguridad informática.
d. Integración de módulos.

Pregunta 13

¿Qué se evalúa en las pruebas de rendimiento del software?

Opciones:

a. La capacidad del software para operar bajo carga.
b. La exactitud de los cálculos realizados por el software.
c. La facilidad de instalación y configuración del software.
d. La coherencia estética de la interfaz de usuario.

Pregunta 14

Si durante las pruebas de integración se descubre que dos módulos no se comunican correctamente, ¿cuál sería el siguiente paso más apropiado?

Opciones:

a. Cambiar los requerimientos para que los módulos no necesiten interactuar.
b. Eliminar uno de los módulos problemáticos.
c. Continuar con otras pruebas ignorando el problema.
d. Revisar y corregir la interfaz de comunicación entre los módulos.

Pregunta 15

Si durante las pruebas se identifica un error crítico en la seguridad, ¿cuál sería el procedimiento adecuado?

Opciones:

a. Suspender todas las pruebas hasta que se resuelva el error.
b. Reportar el error y priorizar su corrección antes de continuar.
c. Continuar las pruebas asumiendo que el error se solucionará en el futuro.
d. Cambiar el alcance del proyecto para excluir la funcionalidad afectada.

Pregunta 16

¿Cuál es la diferencia principal entre un error y un defecto en el testing de software?

Opciones:

a. Un error es una acción humana incorrecta, mientras que un defecto es un problema en el código resultante.
b. Un error y un defecto son lo mismo.
c. Un error es un problema en el código, mientras que un defecto es un fallo en el sistema.
d. Un defecto es un error en el diseño, mientras que un error es un fallo en la implementación.

Pregunta 17

¿Cuál es el propósito de documentar un defecto en el desarrollo de software?

Opciones:

a. Para mantener un registro de todos los errores cometidos durante el desarrollo.
b. Para penalizar al desarrollador que cometió el error.
c. Para proporcionar información detallada para que el defecto pueda ser corregido.
d. Para mantener un registro de todos los errores cometidos durante el desarrollo.

Pregunta 18

¿Cuál es el impacto de no gestionar correctamente los defectos en el desarrollo de software?

Opciones:

a. Mejora en la eficiencia del equipo de desarrollo.
b. Aumento en los costos de desarrollo y posible pérdida de confianza del cliente.
c. Reducción en el tiempo necesario para el desarrollo del software.
d. Ninguno, ya que los defectos siempre se encuentran eventualmente.

Pregunta 19

Durante el testing de software, si un defecto es identificado, pero no afecta significativamente la funcionalidad, ¿cuál sería la acción más apropiada?

Opciones:

a. Informar al cliente y pedir una extensión del plazo de entrega.
b. Detener inmediatamente todo el desarrollo hasta que se resuelva el defecto.
c. Ignorar el defecto y continuar con el desarrollo.
d. Asignar baja prioridad y planificar su resolución en futuras actualizaciones.

Pregunta 20

Si un defecto identificado es crítico para la funcionalidad del software, ¿cuál debería ser el enfoque del equipo de desarrollo?

Opciones:

a. Esperar a que el defecto se resuelva por sí solo.
b. Asignar el defecto a un miembro junior del equipo para su aprendizaje.
c. Priorizar la resolución del defecto antes de continuar con otras tareas.
d. Cambiar los requisitos del software para acomodar el defecto.

Pregunta 21

¿Qué objetivo tienen las técnicas de pruebas estáticas en el desarrollo de software?

Opciones:

a. Detectar y corregir defectos tempranamente en el proceso de desarrollo.
b. Evaluar el rendimiento del software después de su implementación.
c. Realizar pruebas de usabilidad y experiencia de usuario.
d. Validar la funcionalidad del software en su entorno final de operación.

Pregunta 22

¿Cuál es la principal diferencia entre las pruebas de caja negra y las pruebas de caja blanca?

Opciones:

a. La caja negra se enfoca en la funcionalidad externa y la caja blanca en la estructura interna del código.
b. La caja negra prueba la interfaz de usuario, mientras que la caja blanca prueba la base de datos.

c. La caja negra requiere conocimientos de programación, mientras que la caja blanca no.
d. La caja negra es para pruebas de rendimiento y la caja blanca para pruebas de seguridad.

Pregunta 23

¿Qué se busca con las pruebas basadas en la experiencia?

Opciones:

a. Automatizar completamente el proceso de pruebas.
b. Utilizar la intuición y experiencia previa del probador para identificar posibles defectos.
c. Cumplir estrictamente con los requisitos formales de prueba.
d. Seguir un conjunto riguroso y predefinido de pruebas.

Pregunta 24

En un desarrollo de videojuego con múltiples niveles, ¿cuál técnica de prueba sería más adecuada para asegurar la funcionalidad antes de avanzar al siguiente nivel?

Opciones:

a. Pruebas basadas en la experiencia.
b. Pruebas de caja blanca.
c. Análisis de valores límite.
d. Pruebas de regresión.

Pregunta 25

Si estás desarrollando un módulo de registro de pacientes y necesitas garantizar la precisión de los datos ingresados, ¿qué técnica de prueba aplicarías?

Opciones:

a. Pruebas de caja negra.
b. Partición equivalente.
c. Pruebas de caja blanca.
d. Pruebas exploratorias.

Pregunta 26

¿Qué implica el enfoque Agile en las pruebas de software?

Opciones:

a. Ignorar la documentación y centrarse únicamente en la codificación.
b. Realizar pruebas solo al final del ciclo de desarrollo.
c. Integrar las pruebas como parte continua del proceso de desarrollo.
d. Dejar las pruebas únicamente a los especialistas en testing.

Pregunta 27

¿Cómo se diferencia el Agile Testing de los métodos tradicionales de prueba?

Opciones:

a. Agile Testing es menos estructurado y carece de planificación.
b. El Agile Testing se centra más en la documentación que en las pruebas prácticas.
c. En el Agile Testing las pruebas se realizan solo después de completar todo el desarrollo.
d. Agile Testing implica una colaboración constante y pruebas integradas en todo el ciclo de desarrollo.

Pregunta 28

¿Qué beneficio clave ofrece el Test-Driven Development (TDD) en Agile Testing?

Opciones:

a. Se centra únicamente en las pruebas de rendimiento del software.
b. Elimina la necesidad de realizar pruebas manuales.
c. Permite escribir menos código y reducir el tiempo de pruebas.
d. Facilita la detección y corrección temprana de defectos a través de un ciclo iterativo de desarrollo y pruebas.

Pregunta 29

Si estás implementando pruebas en un proyecto Agile que utiliza sprints, ¿cómo aplicarías las pruebas de sistema para maximizar la eficiencia?

Opciones:

a. Integrar pruebas de sistema en cada sprint para evaluar la aplicación completa.
b. Realizar las pruebas de sistema solo al final de todos los sprints.
c. Limitar las pruebas de sistema a los primeros sprints.
d. Evitar las pruebas de sistema en proyectos Agile debido a limitaciones de tiempo.

Pregunta 30

Al trabajar en un proyecto de desarrollo de software con metodologías Agile, ¿cómo se integrarían mejor las pruebas de aceptación del usuario?

Opciones:

a. Omitirlas en favor de pruebas automáticas.
b. Delegarlas exclusivamente a los usuarios finales sin participación del equipo de desarrollo.
c. Realizarlas solo al final del desarrollo del proyecto.
d. Llevarlas a cabo de manera continua a lo largo del proyecto.

Pregunta 31

¿Cuál es la importancia de gestionar los riesgos en el área de calidad de software?

Opciones:

a. Anticiparse a problemas y mitigarlos para asegurar la calidad y eficiencia del software.
b. Reducir los costos de desarrollo al eliminar las pruebas.
c. Garantizar que no existan errores o fallas en el software.
d. Acelerar el lanzamiento del software ignorando posibles riesgos.

Pregunta 32

¿Qué se entiende por "Riesgo RBT" en el testing de software?

Opciones:

a. Los riesgos identificados y evaluados mediante el método de pruebas basadas en riesgos.
b. El riesgo asociado con el uso de herramientas de prueba automatizadas.
c. Riesgos asociados exclusivamente con pruebas de regresión.
d. Riesgos que no pueden ser mitigados o gestionados.

Pregunta 33

¿Cómo se definen los riesgos en el testing de software?

Opciones:

a. Como problemas que siempre llevan a fallas críticas del software.
b. Como factores capaces de producir consecuencias negativas, evaluados por su probabilidad e impacto.
c. Como errores menores que no afectan la funcionalidad general del software.
d. Como desafíos técnicos que solo pueden ser abordados por expertos en seguridad.

Pregunta 34

En un proyecto de desarrollo de software Agile, ¿cómo se debería abordar la gestión de riesgos para maximizar la eficiencia?

Opciones:

a. Dejando la gestión de riesgos hasta el final del proyecto.
b. Ignorando los riesgos menores para centrarse en el desarrollo.
c. Identificando y evaluando los riesgos al inicio de cada iteración o sprint.
d. Externalizando la gestión de riesgos a un equipo independiente.

Pregunta 35

¿Cómo pueden los informes de prueba ayudar en la gestión de riesgos en proyectos de software Agile?

Opciones:

a. Ofreciendo una visión general de los riesgos y el progreso de las pruebas.
b. Proporcionando una descripción detallada de cada línea de código.
c. Centrándose únicamente en los resultados de pruebas automatizadas.
d. Documentando solo los defectos críticos encontrados.

Pregunta 36

¿Cómo se benefician los proyectos Agile de la implementación de pruebas exploratorias?

Opciones:

a. Eliminan la necesidad de pruebas en etapas posteriores del desarrollo.
b. Reducen la necesidad de comunicación entre el equipo de desarrollo y el de pruebas.
c. Permiten una planificación detallada y exhaustiva de pruebas.
d. Facilitan la detección temprana de defectos y mejoran la calidad del software.

Pregunta 37

¿Qué elemento NO es esencial en una bitácora de resultados de pruebas Agile?

Opciones:

a. Diagramas detallados de diseño de software.
b. Fecha de ejecución de la prueba.
c. Estado de la prueba (aprobada o fallida).
d. Resultados obtenidos.

Pregunta 38

¿Qué característica de las pruebas exploratorias las hace especialmente adecuadas para el testing Agile?

Opciones:

a. Se enfocan exclusivamente en la funcionalidad del software.
b. No requieren documentación ni seguimiento.
c. Se basan en pruebas completamente automatizadas.
d. Dependen del juicio y habilidad del equipo de pruebas.

Pregunta 39

¿Qué característica es clave para un examinador de pruebas exploratorias?

Opciones:

a. Tener habilidades avanzadas en programación.
b. Ser un experto en herramientas de automatización de pruebas.
c. Ser altamente experimentado en desarrollo de software.
d. Poseer un conocimiento detallado de los requisitos del cliente.

Pregunta 40

¿Cómo se documentan mejor los resultados de las pruebas exploratorias en Agile Testing?

Opciones:

a. Solo mediante herramientas de seguimiento automatizadas.
b. A través de informes formales detallados.
c. Utilizando una bitácora de pruebas rápida y dinámica.
d. No se requiere documentación para este tipo de pruebas.