

Pregunta #A6 (1 punto)

¿Cuál de las siguientes opciones explica MEJOR un beneficio de la independencia de las pruebas?

- a) El uso de un equipo de prueba independiente permite a la dirección del proyecto asignar la responsabilidad de la calidad del producto final al equipo de prueba. b) Si se puede permitir un equipo de prueba externo a la organización, entonces existen claros beneficios en términos de que este equipo externo no se deje influenciar tan fácilmente por las preocupaciones de entrega de la gestión de proyectos y la necesidad de cumplir plazos de entrega estrictos. c) Un equipo de pruebas independiente puede trabajar por separado de los desarrolladores, no necesita distraerse con los cambios en los requisitos del proyecto y puede restringir la comunicación con los desarrolladores para informar sobre defectos a través del sistema de gestión de defectos. d) Cuando las especificaciones contienen ambigüedades e inconsistencias, se hacen suposiciones sobre su interpretación, y un evaluador independiente puede ser útil para cuestionar esas suposiciones y la interpretación hecha por el desarrollador.

Seleccione UNA opción.

Pregunta #A7 (1 punto)

Trabajas como tester en el equipo que sigue el modelo V. ¿Cómo afecta la elección de este modelo de ciclo de vida de desarrollo de software (SDLC) al momento de las pruebas?

- a) Las pruebas dinámicas no se pueden realizar al principio del SDLC. b) Las pruebas estáticas no se pueden realizar al principio del SDLC. c) La planificación de pruebas no se puede realizar en una fase temprana del SDLC. d) Las pruebas de aceptación se pueden realizar al principio del SDLC.

Seleccione UNA opción.

Pregunta #A8 (1 punto)

¿Cuáles de las siguientes son ventajas de DevOps?

- i. Lanzamiento de producto más rápido y tiempo de comercialización más rápido. ii. Aumenta la necesidad de realizar pruebas manuales repetitivas. III. Disponibilidad constante de software ejecutable. IV. Reducción del número de pruebas de regresión asociadas a la refactorización de código. v. Configurar el marco de automatización de pruebas es económico ya que todo está automatizado.

- a) i, ii y iv son ventajas; iii y v no lo son. b) Iii yv son ventajas; i, ii y iv no lo son. c) i y iii son ventajas; ii, iv y v no lo son. d) ii, iv y v son ventajas; I y III no lo son.

Seleccione UNA opción.

Pregunta #A9 (1 punto)

Trabajas como tester en un proyecto sobre una aplicación móvil para pedidos de comida para uno de tus clientes. El cliente le envió una lista de requisitos. Uno de ellos, con alta prioridad, dice

El pedido debe procesarse en menos de 10 segundos en el 95% de los casos.

Creó un conjunto de casos de prueba en los que se realizaron varios pedidos aleatorios, se midió el tiempo de procesamiento y los resultados de las pruebas se compararon con los requisitos.

¿Qué tipo de prueba realizó?

- a) Funcional, porque los casos de prueba cubren los requisitos comerciales del usuario para el sistema.
- b) No funcionales, porque miden el desempeño del sistema.
- c) Funcional, porque los casos de prueba interactúan con la interfaz de usuario.
- d) Estructural, porque necesitamos conocer la estructura interna del programa para medir el tiempo de procesamiento de pedidos.

Seleccione UNA opción.

Pregunta #A10 (1 punto)

La estrategia de prueba de su organización sugiere que una vez que se retire un sistema, se debe probar la migración de datos. ¿Como parte de qué tipo de prueba es MÁS probable que se realice esta prueba?

- a) Pruebas de mantenimiento.
- b) Pruebas de regresión.
- c) Pruebas de componentes.
- d) Pruebas de integración.

Seleccione UNA opción.

Pregunta #A11 (1 punto)

La siguiente es una lista de los productos de trabajo producidos en el SDLC.

- i. Requisitos comerciales.
- ii. Cronograma.

III. Presupuesto de

prueba.

- IV. Código ejecutable de terceros.
- v. Historias de usuarios y sus criterios de aceptación.

¿Cuáles de ellos se pueden revisar?

- a) i y iv pueden revisarse; ii, iii y v no pueden.
- b) se pueden revisar i, ii, iii y iv; No puedo.
- c) se pueden revisar i, ii, iii y v; No puedo.
- d) iii, iv y v pueden ser revisados; I y II no pueden.

Seleccione UNA opción.

Pregunta #A12 (1 punto)

Decida cuáles de las siguientes afirmaciones (i–v) son verdaderas para las pruebas dinámicas y cuáles son verdaderas para las pruebas estáticas.

- i. Los comportamientos externos anormales son más fáciles de identificar con esta prueba. ii. Las discrepancias con respecto a un estándar de codificación son más fáciles de encontrar con esta prueba. III. Identifica fallas causadas por defectos cuando se ejecuta el software. IV. Su objetivo de prueba es identificar defectos lo antes posible. v. La cobertura faltante para requisitos de seguridad críticos es más fácil de encontrar y solucionar.
- a) i, iv y v son verdaderos para pruebas estáticas; ii y iii son ciertos para las pruebas dinámicas. b) i, iii y iv son verdaderos para pruebas estáticas; ii y v son ciertos para pruebas dinámicas. c) ii y iii son ciertos para pruebas estáticas; i, iv y v son verdaderos para pruebas dinámicas. d) ii, iv y v son verdaderos para pruebas estáticas; i, iii y iv son válidos para pruebas dinámicas.

Seleccione UNA opción.

Pregunta #A13 (1 punto)

¿Cuál de las siguientes afirmaciones sobre revisiones formales es VERDADERA?

- a) Algunas revisiones no requieren más de un rol. b) El proceso de revisión tiene varias actividades. c) La documentación a revisar no se distribuye antes de la reunión de revisión, con excepción del producto del trabajo para tipos de revisión específicos. d) Los defectos encontrados durante la revisión no se reportan ya que no se encuentran mediante pruebas dinámicas.

Seleccione UNA opción.

Pregunta #A14 (1 punto)

¿Qué tarea puede asumir la dirección durante una revisión formal?

- a) Asumir la responsabilidad general de la revisión. b) Decidir qué se va a revisar. c) Velar por el eficaz desarrollo de las reuniones de revisión y mediar, en su caso. d) Registrar información de revisión, como decisiones de revisión.

Seleccione UNA opción.

Pregunta #A15 (1 punto)

Un sistema de almacenamiento de vino utiliza un dispositivo de control que mide la temperatura de la celda del vino. T (medido en °C, redondeado al grado más cercano) y alerta al usuario si se desvía del valor óptimo de 12, de acuerdo con las siguientes reglas:

- Si $T = 12$, el sistema dice "Temperatura óptima".
- Si $T < 12$, el sistema dice: "¡La temperatura es demasiado baja!"
- Si $T > 12$, el sistema dice: "¡La temperatura es demasiado alta!"

Desea utilizar el análisis de valor límite (BVA) de 3 valores para verificar el comportamiento del dispositivo de control. Una entrada de prueba es una temperatura en °C proporcionada por el dispositivo.

¿Cuál es el conjunto MÍNIMO de entradas de prueba que logra el 100% de la cobertura deseada?

- a) 11, 12, 13 b) 10,
12, 14 c) 10, 11,
12, 13, 14 d) 10, 11, 13, 14

Seleccione UNA opción.

Pregunta #A16 (1 punto)

¿Cuál de las siguientes afirmaciones sobre las pruebas de rama es CORRECTA?

- a) Si un programa incluye solo sucursales incondicionales, entonces se puede lograr una cobertura de sucursales del 100% sin ejecutar ningún caso de prueba. b) Si los casos de prueba ejercen todas las ramas incondicionales del código, entonces 100% Se logra cobertura de sucursales.
- c) Si se logra una cobertura del 100% del estado de cuenta, entonces también se alcanza una cobertura del 100% de la sucursal. logrado.
- d) Si se logra una cobertura del 100% de las sucursales, entonces todos los resultados de las decisiones en cada se ejercen las declaraciones de decisión contenidas en el código.

Seleccione UNA opción.

Pregunta #A17 (1 punto)

Está probando una aplicación móvil que permite a los clientes acceder y administrar sus cuentas bancarias. Está ejecutando un conjunto de pruebas que implica evaluar cada pantalla y cada campo de cada pantalla con una lista general de mejores prácticas de interfaz de usuario derivadas de un libro popular sobre el tema que maximiza el atractivo, la facilidad de uso y la accesibilidad para dichas aplicaciones.

¿Cuál de las siguientes opciones categoriza MEJOR la técnica de prueba que está utilizando?

- a) Caja negra. b) Exploratorio. c) Basado en listas de verificación. d) Error al adivinar.

Seleccione UNA opción.

Pregunta #A18 (1 punto)

¿Cuál de las siguientes opciones describe MEJOR el enfoque colaborativo para la redacción de historias de usuario?

- a) Las historias de usuario son creadas por evaluadores y desarrolladores y luego aceptadas por representantes comerciales. b) Las historias de usuario son creadas por representantes comerciales, desarrolladores y evaluadores. juntos.
- c) Las historias de usuario son creadas por representantes comerciales y verificadas por desarrolladores. y probadores.

d) Las historias de usuario se crean de manera que sean independientes, negociables, valiosas, estimable, pequeño y comprobable.

Seleccione UNA opción.

Pregunta #A19 (1 punto)

Considere la siguiente parte de un plan de prueba.

Las pruebas se realizarán mediante pruebas de componentes e integración de componentes. pruebas. La normativa exige demostrar que el 100% de cobertura de sucursales es logrados para cada componente clasificado como crítico.

¿A qué parte del plan de prueba pertenece esta parte?

- a) Comunicación.
- b) Registro de riesgos.
- c) Contexto de las pruebas.
- d) Enfoque de prueba.

Seleccione UNA opción.

Pregunta #A20 (1 punto)

Su equipo utiliza el póquer de planificación para estimar el esfuerzo de prueba para un nuevo requisito. característica. Hay una regla en su equipo que dice que si no hay tiempo para llegar a un acuerdo total y la variación en los resultados es pequeña, aplicando reglas como "aceptar el número con se puede aplicar la mayor cantidad de votos".

Después de dos rondas, no se alcanzó consenso, por lo que se inició la tercera ronda. Puede ver los resultados de la estimación de la prueba en la siguiente tabla.

	Estimaciones de los miembros del equipo						
La ronda 1	21	2	5	34	13	8	2
La ronda 2	13	8	8	34	13	8	5
Ronda 3	13	8	13	13	13	13	8

¿Cuál de los siguientes es el MEJOR ejemplo del siguiente paso?

- a) El propietario del producto debe intervenir y tomar una decisión final.
- b) Acepte 13 como estimación de prueba final, ya que tiene la mayoría de los votos.
- c) No es necesaria ninguna otra acción. Se ha llegado a un consenso.
- d) Eliminar la nueva característica de la versión actual porque no se ha logrado consenso. sido alcanzado.

Seleccione UNA opción.

Pregunta #A21 (1 punto)

¿Cuál de las siguientes afirmaciones NO es cierta con respecto a la pirámide de prueba?

- a) La pirámide de pruebas enfatiza tener una mayor cantidad de pruebas en la prueba inferior niveles.
- b) Cuanto más cerca de la cima de la pirámide, más formal será la automatización de sus pruebas. debiera ser.

c) Por lo general, las pruebas de componentes y las pruebas de integración de componentes se automatizan utilizando herramientas basadas en API. d) Para las pruebas del sistema y las pruebas de aceptación, las pruebas automatizadas generalmente se crean utilizando herramientas basadas en GUI.

Seleccione UNA opción.

Pregunta #A22 (1 punto)

Durante el análisis de riesgos, el equipo consideró el siguiente riesgo: El sistema permite un descuento demasiado alto para un cliente. El equipo estimó que el impacto del riesgo era muy alto.

¿Qué se puede decir sobre la probabilidad de riesgo?

a) También es muy alto. Un impacto de alto riesgo siempre implica una alta probabilidad de riesgo. b) Es muy bajo. Un impacto de alto riesgo siempre implica una baja probabilidad de riesgo. c) No se puede decir nada sobre la probabilidad del riesgo. El impacto del riesgo y la probabilidad del riesgo son independientes. d) La probabilidad del riesgo no es importante con un impacto de riesgo tan alto. uno no necesita para definirlo.

Seleccione UNA opción.

Pregunta #A23 (1 punto)

La siguiente lista contiene riesgos que se han identificado para el desarrollo de un nuevo producto de software:

i. La dirección traslada a dos evaluadores experimentados a otro proyecto. ii. El sistema no cumple con los estándares de seguridad funcional. III. El tiempo de respuesta del sistema excede los requisitos del usuario. IV. Las partes interesadas tienen expectativas inexactas. v. Las personas con discapacidad tienen problemas al utilizar el sistema.

¿Cuáles de ellos son riesgos del proyecto?

a) i y iv son riesgos del proyecto; ii, iii y v no son riesgos del proyecto. b) iv y v son riesgos del proyecto; i, ii y iii no son riesgos del proyecto. c) i y iii son riesgos del proyecto; ii, iv y v no son riesgos del proyecto. d) ii y v son riesgos del proyecto; i, iii y iv no son riesgos del proyecto.

Seleccione UNA opción.

Pregunta #A24 (1 punto)

¿Cuál de los siguientes es un ejemplo de cómo el análisis de riesgos del producto influye en la minuciosidad y el alcance de las pruebas?

a) El director de pruebas monitorea e informa diariamente el nivel de todos los riesgos conocidos para que las partes interesadas puedan tomar una decisión informada sobre la fecha de lanzamiento. b) Uno de los riesgos identificados fue la "Falta de soporte de bases de datos de código abierto", por lo que el equipo decidió integrar el sistema con una base de datos de código abierto. c) Durante el análisis de riesgo cuantitativo, el equipo estimó el nivel total de todos los riesgos identificados y lo informó como el riesgo residual total antes de la prueba.

d) La evaluación de riesgos reveló un nivel muy alto de riesgos de desempeño, por lo que se decidió realizar pruebas detalladas de eficiencia del desempeño al principio del SDLC.

Seleccione UNA opción.

Pregunta #A25 (1 punto)

¿Cuáles DOS de las siguientes opciones son métricas comunes utilizadas para informar sobre el nivel de calidad del objeto de prueba?

- a) Número de defectos encontrados durante las pruebas del sistema.
- b) Esfuerzo total en el diseño de pruebas dividido por el número de casos de prueba diseñados. c) Número de procedimientos de prueba ejecutados. d) Número de defectos encontrados dividido por el tamaño de un producto de trabajo. e) Tiempo necesario para reparar un defecto.

Seleccione DOS opciones.

Pregunta #A26 (1 punto)

¿Cuál de los siguientes datos contenidos en un informe de progreso de prueba es MENOS útil para los representantes comerciales?

- a) Impedimentos para realizar las pruebas. b) Cobertura de sucursales lograda. c) Progreso de la prueba. d) Nuevos riesgos dentro del ciclo de pruebas.

Seleccione UNA opción.

Conjunto de exámenes A: Respuestas



Pregunta #1

FL-4.2.2 (K3)

Respuesta correcta: a

Hay 12 valores límite para los valores del resultado final: 0, 50, 51, 60, 61, 70, 71, 80, 81, 90, 91 y 100. Los casos de prueba cubren seis de ellos (TC1, 91; TC2, 50; TC3, 81; Por lo tanto, los casos de prueba cubren 6/12 = 50%.

a) Es correcto. b)

No es correcto. c) No es

correcto. d) No es

correcto.

Pregunta #2

FL-3.2.1 (K1)

Respuesta correcta: d.

a) No es correcto. La retroalimentación puede mejorar el proceso de prueba, pero si uno solo quiere mejorar proyectos futuros, no es necesario que la retroalimentación llegue temprano o con frecuencia.

b) No es correcto. La retroalimentación no se utiliza para priorizar los requisitos. c) No es correcto. La calidad de los cambios se puede medir de múltiples maneras. d) Es correcto. La retroalimentación temprana y frecuente permite la comunicación temprana de posibles problemas de calidad.

Pregunta #3

FL-4.2.3 (K3)

Respuesta correcta: d.

a) No es correcto. Un miembro que no haya incumplido el plazo puede obtener un descuento y una camiseta de regalo después de 15 alquileres de

bicicletas. b) No es correcto. Un miembro sin una fecha límite incumplida puede obtener un descuento pero ningún regalo. Camiseta hasta que alquilaron una bicicleta 15 veces.

- c) No es correcto. Los no miembros no pueden obtener un descuento, incluso si no se perdieron un fecha límite todavía.
- d) Es correcto. No habrá descuento como no socio que además haya incumplido un plazo, pero sólo los socios podrán recibir una camiseta de regalo. Por tanto, la acción no es correcta.

Pregunta #4

FL-5.4.1 (K2)

Respuesta correcta:c

- a) No es correcto. La trazabilidad es la relación entre dos o más productos de trabajo, no entre diferentes versiones del mismo producto de trabajo. b) No es correcto. Las pruebas de mantenimiento consisten en probar cambios; no está estrechamente relacionado con el control de versiones.
- c) Es correcto. Para respaldar las pruebas, la gestión de la configuración puede implicar el control de versiones de todos los elementos de prueba.
- d) No es correcto. La ingeniería de requisitos es la obtención, documentación y gestión de requisitos; no está estrechamente relacionado con el control de versiones del script de prueba.

Pregunta #5

FL-4.1.1 (K2)

Respuesta correcta:c

- a) No es correcto. Ésta es una característica común de las técnicas de prueba de caja blanca. Las condiciones de prueba, los casos de prueba y los datos de prueba se derivan de una base de prueba que puede incluir código, arquitectura de software, diseño detallado o cualquier otra fuente de información sobre la estructura del software.
- b) No es correcto. Ésta es una característica común de las técnicas de prueba de caja blanca. La cobertura se mide en función de los elementos probados dentro de una estructura seleccionada y la técnica aplicada a la base de la prueba. c) Es correcto. Esta es una característica común de las técnicas de prueba basadas en la experiencia. Este conocimiento y experiencia incluyen el uso esperado del software, su entorno, defectos probables y la distribución de aquellos defectos utilizados para definir las pruebas. d) No es correcto. Ésta es una característica común de las técnicas de prueba de caja negra. Los casos de prueba se pueden utilizar para detectar brechas dentro de los requisitos y la implementación de los mismos, así como desviaciones de los requisitos.

Pregunta #6

FL-4.2.4 (K3)

Respuesta correcta: d

Las transiciones de "prueba" y "error" no pueden ocurrir en un caso de prueba. Tampoco pueden ambas transiciones "terminadas". Esto significa que necesitamos al menos tres casos de prueba para lograr la cobertura de transición. Por ejemplo:

TC1: prueba, hecho

TC2: ejecutar, error, hecho

TC3: ejecutar, pausar, reanudar, pausar, listo

Por eso

- a) No es correcto b) No es correcto c) No es correcto d) Es correcto

Pregunta #7

FL-5.5.1 (K3)

Respuesta correcta:c

- a) No es correcto. El resultado esperado es "la aplicación debería aceptar la entrada proporcionada y crear el usuario". El resultado real es "La aplicación se cuelga después de ingresar 'Entrada de prueba'. \$ä."
- b) No es correcto. Hay una referencia al caso de prueba y al requisito relacionado, y establece que se rechaza el defecto. Además, el estado del defecto no sería de mucha ayuda para los desarrolladores.
- c) Es correcto. No sabemos en qué entorno de prueba se detectó la anomalía y tampoco sabemos qué aplicación (y su versión) está afectada. d) No es correcto. El informe de defectos afirma que la anomalía es urgente y que es un problema global (es decir, muchas, si no todas, las cuentas de administración de pruebas se ven afectadas) y afirma que el impacto es alto para las partes interesadas del negocio.

Pregunta #8

FL-5.1.2 (K1)

Respuesta correcta:c

- a) No es correcto. Las prioridades para las historias de usuarios están determinadas por los representantes comerciales. representante junto con el equipo de desarrollo.
- b) No es correcto. Los evaluadores se centran en aspectos funcionales y no funcionales del sistema a probar.
- c) Es correcto. Según el plan de estudios, esta es una de las formas en que los evaluadores agregan valor a Planificación de iteraciones y lanzamientos.
- d) No es correcto. El diseño de prueba inicial no forma parte de la planificación del lanzamiento. El diseño temprano de las pruebas no garantiza automáticamente el lanzamiento de software de calidad.

Pregunta #9

FL-1.4.1 (K2)

Respuesta correcta:b

- a) No es correcto. Estimar el esfuerzo de la prueba es parte de la planificación de la prueba.
- b) Es correcto. Este es un ejemplo de definición de condiciones de prueba, que es parte de la prueba. análisis.
- c) No es correcto. El uso de técnicas de prueba para derivar elementos de cobertura es parte de la prueba. diseño.
- d) No es correcto. Informar los defectos encontrados durante las pruebas dinámicas es parte de la prueba. ejecución.

Pregunta #10

FL-5.3.3 (K2)

Respuesta correcta: d.

- a) No es correcto. Los criterios de aceptación son las condiciones utilizadas para decidir si el La historia del usuario está lista. No pueden mostrar el progreso del trabajo.
- b) No es correcto. Los informes de defectos informan sobre los defectos. No muestran trabajo progreso.
- c) No es correcto. El informe de finalización de la prueba se puede crear una vez finalizada la iteración, por lo que no mostrará el progreso continuamente dentro de una iteración. d) Es correcto. Los gráficos de evolución son una representación gráfica del trabajo que queda por hacer en comparación con el tiempo restante. Se actualizan diariamente, por lo que pueden mostrar continuamente el avance del trabajo.

Pregunta #11

FL-2.1.3 (K1)

Respuesta correcta:c

- a) No es correcto. Se utiliza con mayor frecuencia en el desarrollo impulsado por el comportamiento (BDD). b) No es correcto. Es la descripción del desarrollo basado en pruebas (TDD). c) Es correcto. En el desarrollo basado en pruebas de aceptación (ATDD), las pruebas se escriben a partir de criterios de aceptación como parte del proceso de diseño. d) No es correcto. Se utiliza en BDD.

Pregunta #12

FL-6.2.1 (K1)

Respuesta correcta:b

- a) No es correcto. La automatización de pruebas no introduce regresiones desconocidas en producción.
- b) Es correcto. La asignación incorrecta de esfuerzos para mantener el software de prueba es un riesgo.
- c) No es correcto. Las herramientas de prueba deben seleccionarse de manera que ellas y su software de prueba puedan ser confiado.
- d) No es correcto. El objetivo principal de la automatización de pruebas es reducir las pruebas manuales. Entonces, esto es un beneficio, no un riesgo.

Pregunta #13

FL-5.1.3 (K2)

Respuesta correcta: c, e

- a) No es correcto. La preparación del entorno de prueba es un criterio de disponibilidad de recursos; por tanto, pertenece a los criterios de entrada. b) No es correcto. Este es un criterio de disponibilidad de recursos; por tanto, pertenece a la criterio para entrar.
- c) Es correcto. La densidad estimada de defectos es una medida de diligencia; por lo tanto, pertenece a los criterios de salida.
- d) No es correcto. Los requisitos traducidos a un formato determinado dan como resultado requisitos comprobables; por tanto, pertenece a los criterios de entrada.

- e) Es correcto. La automatización de las pruebas de regresión es un criterio de finalización; por lo tanto, pertenece a los criterios de salida.

Pregunta #14

FL-3.1.2 (K2)

Respuesta correcta: una

- a) Es correcto. La gestión de defectos no es menos costosa. Encontrar y reparar defectos más adelante en SDLC es más costoso.
b) No es correcto. Este es un beneficio de las pruebas estáticas. c) No es correcto. Este es un beneficio de las pruebas estáticas. d) No es correcto. Este es un beneficio de las pruebas estáticas.

Pregunta #15

FL-1.5.1 (K2)

Respuesta correcta:b

- i. Es verdad. Tener conocimiento del dominio es una habilidad importante del evaluador.
ii. Es falso. Esta es una tarea del analista de negocios junto con el negocio.
representante. III.

Es verdad. Ser un buen jugador de equipo es una habilidad importante. IV. Es

falso. Planificar y organizar el trabajo del equipo es tarea del responsable de pruebas o, sobre todo en un proyecto de desarrollo de software ágil, de todo el equipo y no sólo del tester. v. Es cierto. El pensamiento crítico es una de las habilidades más importantes de los evaluadores.

Por tanto b es correcto.

Pregunta #16

FL-2.1.6 (K2)

Respuesta correcta:c

- a) No es correcto. Las retrospectivas son más útiles para identificar oportunidades de mejora y tienen poca importancia para los clientes. b) No es correcto. Los representantes comerciales no dan comentarios sobre el producto en sí. Por lo tanto, no hay ningún beneficio económico para la organización. c) Es correcto. Las retrospectivas realizadas periódicamente, cuando se realizan actividades de seguimiento adecuadas, son fundamentales para la mejora continua del desarrollo y las pruebas.
d) No es correcto. El coraje y el respeto son valores de Extreme Programming y no están estrechamente relacionados con las retrospectivas.

Pregunta #17

FL-4.4.2 (K2)

Respuesta correcta:c

- a) No es correcto. Este es un nuevo producto. Probablemente aún no tenga una lista de verificación y es posible que no conozca las condiciones de la prueba debido a que faltan requisitos. b) No es correcto. Este es un nuevo producto. Probablemente no tengas suficiente información para realizar conjeturas erróneas correctas.
- c) Es correcto. Las pruebas exploratorias son más útiles cuando hay pocas especificaciones conocidas. ificaciones, y/o hay un cronograma apremiante para las pruebas.
- d) No es correcto. Las pruebas de sucursales llevan mucho tiempo y su gerencia está preguntando sobre algunos resultados de las pruebas ahora. Además, las pruebas de rama no implican conocimiento del dominio.

Pregunta #18

FL-5.1.4 (K3)

Respuesta correcta:

d En la técnica de estimación de tres puntos

$E = \text{optimista} + 2 \times \text{probable} + \text{pesimista} = 6 + 2 \times 4 + 8 = 18$

Por tanto d es correcto.

Pregunta #19

FL-4.3.3 (K2)

Respuesta correcta: d.

- a) No es correcto. La ventaja fundamental de las técnicas de prueba de caja blanca es que durante las pruebas se tiene en cuenta toda la implementación del software. b) No es correcto. Las medidas de cobertura de caja blanca proporcionan una medida objetiva de la cobertura y proporcionan la información necesaria para permitir que se generen pruebas adicionales para aumentar esta cobertura.
- c) No es correcto. Se pueden utilizar técnicas de prueba de caja blanca para realizar revisiones (estática pruebas).
- d) Es correcto. Ésta es la debilidad de las técnicas de prueba de caja blanca. No pueden identificar la implementación faltante porque se basan únicamente en la estructura del objeto de prueba, no en la especificación de requisitos.

Pregunta #20

FL-3.2.4 (K2)

Respuesta correcta:

b Considerando los atributos:

- Existe una función de escribano, especificada para recorridos, revisiones técnicas e inspecciones; por lo tanto, las revisiones que se realizan no pueden ser revisiones informales.
- El propósito es evaluar la calidad; el propósito de evaluar la calidad es uno de los objetivos más importantes de un recorrido.

- La reunión de revisión está dirigida por el autor del producto del trabajo; esto no está permitido para las inspecciones y normalmente no se hace en las revisiones técnicas. Se necesita un moderador en los recorridos y se le permite realizar revisiones informales.
- Los revisores individuales encuentran posibles anomalías durante la preparación: todo tipo de Las revisiones pueden incluir revisores individuales (incluso revisiones informales). • Se produce un informe de revisión: todos los tipos de revisiones pueden producir un informe de revisión. aunque las revisiones informales no requieren documentación.

Por tanto b es correcto.

Pregunta #21

FL-4.2.1 (K3)

Respuesta correcta:

b "Jardín pequeño" y "jardín grande" solo pueden ir con "planta baja", por lo que necesitamos dos casos de prueba con "planta baja", que cubran estas dos particiones "tipo jardín".

Necesitamos dos casos de prueba más para cubrir las otras dos particiones de "piso" y una partición restante "tipo jardín" de "sin jardín". Necesitamos un total de cuatro casos de prueba:

TC1 (planta baja, pequeño jardín).

TC2 (planta baja, amplio jardín).

TC3 (primer piso, sin jardín).

TC4 (segundo piso o superior, sin jardín).

- a) No es correcto. b) Es correcto. c) No es correcto. d) No es correcto.

Pregunta #22

FL-2.1.5 (K2)

Respuesta correcta: d.

- a) No es correcto. La revisión temprana es un ejemplo del enfoque de desplazamiento a la izquierda. b) No es correcto. TDD es un ejemplo del enfoque de desplazamiento a la izquierda. c) No es correcto. Las pruebas no funcionales tempranas son un ejemplo del desplazamiento hacia la izquierda acercarse. d) Es correcto. Los scripts de prueba deben estar sujetos a la gestión de configuración, por lo que no tiene sentido crear los scripts de prueba antes de configurar este proceso.

Pregunta #23

FL-6.1.1 (K2)

Respuesta correcta:c

- a) No es correcto. El monitoreo de pruebas implica la verificación continua de todas las actividades y la comparación del progreso real con el plan de pruebas. El control de pruebas implica tomar las acciones necesarias para cumplir con los objetivos de prueba del plan de pruebas. Durante estas actividades no se preparan datos de prueba. b) No es correcto. El análisis de pruebas incluye analizar la base de la prueba para identificar las condiciones de la prueba y priorizarlas. El diseño de la prueba incluye la elaboración de la prueba.

condiciones en casos de prueba y otro software de prueba. Los datos de prueba no se preparan durante estas actividades.

c) Es correcto. La implementación de la prueba incluye la creación o adquisición del software de prueba.

necesario para la ejecución de la prueba (por ejemplo, datos de prueba).

d) No es correcto. Las actividades de finalización de las pruebas ocurren en los hitos del proyecto (por ejemplo, lanzamiento, final de la iteración, finalización del nivel de prueba), por lo que es demasiado tarde para preparar los datos de la prueba.

Pregunta #24 FL-4.3.1

(K2)

Respuesta correcta: una

a) Es correcto. Dado que se logra una cobertura del 100% de los extractos, todos los extractos, incluidos los defectuosos, deben haber sido ejecutados y evaluados al menos una vez b) No es correcto. La cobertura depende de lo que se prueba, no de la cantidad de casos de prueba. Por ejemplo, para el código "if (x==0) y=1", un caso de prueba (x=0) logra una cobertura de declaración del 100%, pero dos casos de prueba (x=1) y (x=2) juntos solo logran 50% de cobertura del estado de cuenta.

c) No es correcto. Si hay un bucle en el código, puede haber un número infinito de rutas posibles, por lo que no es posible ejecutar todas las rutas posibles en el código. d) No es correcto. No es posible realizar pruebas exhaustivas (consulte la sección de los siete principios de las pruebas en el programa de estudios). Por ejemplo, para el código "entrada x; print x" cualquier prueba con x arbitraria logra una cobertura de declaración del 100%, pero cubre un valor de entrada.

Pregunta #25

FL-2.1.2 (K1)

Respuesta correcta: d.

a) No es correcto. b) No es

correcto. c) No es correcto.

d) Es correcto; Esta regla

es válida para todos los modelos SDLC.

Pregunta #26 FL-5.1.7

(K2)

Respuesta correcta: una

Las pruebas de usabilidad se encuentran en el tercer trimestre (1—C).

La prueba de componentes se encuentra en Q1 (2—A).

Las pruebas funcionales se encuentran en el segundo trimestre (3—B).

Las pruebas de confiabilidad se encuentran en el cuarto trimestre (4—D).

Por tanto a es correcto.

Pregunta #27

FL-2.2.1 (K2)

Respuesta correcta: una

La base de prueba para las pruebas de aceptación son las necesidades comerciales del usuario (1D).

La comunicación entre componentes se prueba durante las pruebas de integración de componentes.

(2B).

Se pueden encontrar fallas en la lógica durante las pruebas de componentes (3A).

Las reglas comerciales son la base de prueba para las pruebas del sistema (4C).

Por tanto a es correcto.

Pregunta #28

FL-4.5.2 (K2)

Respuesta correcta:b

- a) No es correcto. Las retrospectivas se utilizan para capturar lecciones aprendidas y mejorar el proceso de desarrollo y prueba, no para documentar los criterios de aceptación.
- b) Es correcto. Esta es la forma estándar de documentar los criterios de aceptación. c) No es correcto. La comunicación verbal no permite documentar físicamente los criterios de aceptación como parte de una historia de usuario (aspecto "tarjeta" en el modelo 3C). d) No es correcto. Los criterios de aceptación están relacionados con una historia de usuario, no con un plan de prueba. Además, los criterios de aceptación son las condiciones que deben cumplirse para decidir si la historia del usuario está completa. Los riesgos no son tales condiciones.

Pregunta #29

FL-4.4.1 (K2)

Respuesta correcta: una

- a) Es correcto. El concepto básico detrás de la adivinación de errores es que el evaluador intenta adivinar qué errores pudo haber cometido el desarrollador y qué defectos puede haber en el objeto de prueba basándose en experiencias pasadas (y, a veces, en listas de verificación). b) No es correcto. Aunque los evaluadores que solían ser desarrolladores pueden usar su experiencia personal para ayudarlos a adivinar errores, la técnica de prueba no se basa en conocimientos previos de desarrollo. c) No es correcto. La adivinación de errores no es una técnica de usabilidad para adivinar cómo los usuarios

Es posible que no interactúe con el objeto de prueba.

- d) No es correcto. Duplicar la tarea de desarrollo tiene varios defectos que la hacen poco práctica, como que el evaluador tenga habilidades equivalentes a las del desarrollador y el tiempo necesario para realizar el desarrollo. No es un error adivinar.

Pregunta #30

FL-1.4.5 (K2)

Respuesta correcta: a, e

- a) Es correcto. Esto lo hacen los probadores. b) No es correcto. La cartera de pedidos del producto es creada y mantenida por el producto. dueño.
- c) No es correcto. Esto lo hace el equipo de desarrollo.

d) No es correcto. Esta es una función gerencial. e) Es correcto. Esto lo hacen los probadores.

Pregunta #31

FL-5.1.5 (K3)

Respuesta correcta:

primero debe ir un Test TC 001, seguido del TC 002, para satisfacer las dependencias. Luego, TC 003 para satisfacer la prioridad y luego TC 004, seguido por TC 005. Por lo tanto:

a) Es correcto. b) No es correcto. c) No es correcto. d) No es correcto.

Pregunta #32

FL-1.3.1 (K2)

Respuesta correcta: una

- a) Es correcto. Este principio significa que si las mismas pruebas se repiten una y otra vez, eventualmente estas pruebas ya no encontrarán ningún defecto nuevo. Probablemente esta sea la razón por la que todas las pruebas también pasaron en esta versión.
- b) No es correcto. Este principio habla de la creencia errónea de que simplemente encontrar y corregir una gran cantidad de defectos garantizará el éxito de un sistema. c) No es correcto. Este principio dice que una pequeña cantidad de componentes generalmente contienen la mayoría de los defectos.
- d) No es correcto. Este principio establece que probar todas las combinaciones de entradas y condiciones previas no es factible.

Pregunta #33

FL-5.2.4 (K2)

Respuesta correcta:c

- a) No es correcto. No aceptamos el riesgo; Se proponen acciones concretas. b) No es correcto. No se proponen planes de contingencia. c) Es correcto. Las acciones propuestas están relacionadas con las pruebas, que son una forma de riesgo. mitigación. d) No es correcto. El riesgo no se transfiere sino que se mitiga.

Pregunta #34

FL-4.5.3 (K3)

Respuesta correcta: una

- a) Es correcto. Esta prueba cubre dos criterios de aceptación: uno sobre la edición del documento y otro sobre guardar cambios.
- b) No es correcto. Los criterios de aceptación cubren las actividades del editor, no las actividades del propietario del contenido. c) No es correcto. Programar el contenido editado para su publicación puede ser una buena idea. característica, pero no está cubierta por los criterios de aceptación.

- d) No es correcto. Los criterios de aceptación establecen la reasignación de un editor al propietario del contenido, no a otro editor.

Pregunta #35

FL-2.2.3 (K2)

Respuesta correcta: b

Debido a que TC1 y TC3 fallaron en la Ejecución 1 [es decir, prueba (1) y prueba (3)], prueba (4) y

La prueba (6) son pruebas de confirmación.

Debido a que TC2 y TC3 fallaron en la Ejecución 2 [es decir, pruebas (5) y (6)], prueba (8) y prueba (9) también son pruebas de confirmación.

TC2 pasó en la Ejecución 1 [es decir, prueba (2)], por lo que la prueba (5) es una prueba de regresión.

TC1 pasó en la Ejecución 2 [es decir, prueba (4)], por lo que la prueba (7) también es una prueba de regresión.

Por tanto b es correcto.

Pregunta #36

FL-1.5.2 (K1)

Respuesta correcta: d.

- a) No es correcto. El enfoque de automatización de pruebas lo definen los evaluadores con la ayuda de desarrolladores y representantes comerciales.
- b) No es correcto. La estrategia de prueba se decide en colaboración con los desarrolladores. c) No es correcto. Los evaluadores, desarrolladores y representantes comerciales forman parte de este enfoque de equipo completo. d) Es correcto. Los evaluadores trabajarán en estrecha colaboración con representantes comerciales para garantizar que se alcancen los niveles de calidad deseados. Esto incluye apoyarlos y colaborar con ellos para ayudarlos a crear pruebas de aceptación adecuadas.

Pregunta #37

FL-3.2.5 (K1)

Respuesta correcta: d.

- a) No es correcto. El tiempo adecuado para las personas es un factor de éxito. b) No es correcto. Dividir los productos de trabajo en pequeñas partes adecuadas es un éxito factor.
- c) No es correcto. Evitar conductas que puedan indicar aburrimiento, exasperación, etc. es un factor de éxito. d) Es correcto. Durante las revisiones se pueden encontrar defectos, no fallos.

Pregunta #38

FL-1.2.1 (K2)

Respuesta correcta: una

- a) Es correcto. Es importante que los evaluadores participen desde el comienzo del ciclo de vida de desarrollo de software (SDLC). Aumentará la comprensión de las decisiones de diseño y detectará defectos tempranamente.
- b) No es correcto. Tanto los desarrolladores como los evaluadores comprenderán mejor cada uno de ellos. productos de trabajo de otros y cómo probar el código.

- c) No es correcto. Si los evaluadores pueden trabajar estrechamente con los diseñadores de sistemas, les dará información sobre cómo realizar la prueba.
- d) No es correcto. Las pruebas no tendrán éxito si no se comprueba el cumplimiento de los requisitos legales.

Pregunta #39

FL-1.1.1 (K1)

Respuesta correcta:c

- a) No es correcto. Es imposible probar que ya no existen defectos en el sistema bajo prueba. Consulte el principio de prueba 1.
- b) No es correcto. Ver principio de prueba 7. c) Es correcto. Las pruebas encuentran defectos y fallas, lo que reduce el nivel de riesgo y al mismo tiempo brinda más confianza en el nivel de calidad del objeto de prueba. d) No es correcto. Es imposible probar todas las combinaciones de entradas (ver prueba principio 2).

Pregunta #40

FL-1.4.2 (K2)

Respuesta correcta:b

- i. Es verdad. El SDLC influye en el proceso de prueba. ii. Es falso. El número de defectos detectados en proyectos anteriores puede tener alguna influencia, pero esto no es tan significativo como i, iii y iv.
- III. Es verdad. Los riesgos del producto identificados son uno de los factores más importantes. influyendo en el proceso de prueba.
- IV. Es verdad. Los requisitos reglamentarios son factores importantes que influyen en la prueba. proceso.
- v. Es falso. El entorno de prueba debe ser una copia del entorno de producción, pero no tiene una influencia significativa en el proceso de prueba.
- Por tanto b es correcto.

Preguntas de muestra adicionales: respuestas



Pregunta #A1

FL-1.1.2 (K2)

Respuesta correcta: una

- a) Es correcto. La depuración es el proceso de encontrar, analizar y eliminar los Causas de fallas en un componente o sistema.
- b) No es correcto. Las pruebas son el proceso relacionado con la planificación, preparación y evaluación de un componente o sistema y productos de trabajo relacionados para determinar que satisfacen requisitos específicos, demostrar que son aptos para su propósito y detectar defectos. No está relacionado con solucionar las causas de las fallas. c) No es correcto. La obtención de requisitos es el proceso de recopilar, capturar y consolidar requisitos de fuentes disponibles. No está relacionado con solucionar las causas de las fallas.
- d) No es correcto. La gestión de defectos es el proceso de reconocer, registrar, clasificar, investigar, resolver y eliminar defectos. No está relacionado con solucionar las causas de las fallas.

Pregunta #A2

FL-1.2.2 (K1)

Respuesta correcta: d.

- a) No es correcto. Ver justificación d. b) No es correcto. Ver justificación d. c) No es correcto. Ver justificación d. d) Es correcto. Las pruebas y el control de calidad no son lo mismo. Las pruebas son el proceso que consta de todas las actividades del ciclo de vida de desarrollo de software (SDLC), tanto estáticas como dinámicas, relacionadas con la planificación, preparación y evaluación de un componente o sistema y productos de trabajo relacionados para determinar que satisfacen requisitos específicos, para demostrar que son aptos para su propósito y para detectar defectos. El aseguramiento de la calidad se centra en establecer, introducir, monitorear, mejorar y adherirse a los procesos relacionados con la calidad.

Pregunta #A3

FL-1.2.3 (K2)

Respuesta correcta: d.

- a) No es correcto. La causa principal es la distracción que experimentó el programador mientras programaba.
b) No es correcto. Aceptar entradas no válidas es un fracaso. c) No es correcto. El error es el pensamiento erróneo que resultó en poner el defecto en el código. d)

Es correcto. El problema en el código es un defecto.

Pregunta #A4

FL-1.4.3 (K2)

Respuesta correcta: d

El software de prueba considerado es una carta de prueba. Las cartas de prueba son el resultado del diseño de la prueba. Por tanto d es correcto.

Pregunta #A5

FL-1.4.4 (K2)

Respuesta correcta:c

- a) No es correcto. La realización del análisis de impacto no proporcionará información sobre la integridad de las pruebas. Analizar el análisis de impacto de los cambios ayudará a seleccionar los casos de prueba adecuados para su ejecución.
b) No es correcto. La trazabilidad no proporciona información sobre el nivel estimado de riesgo residual si los casos de prueba no se remontan a riesgos.
c) Es correcto. Realizar el análisis de impacto de los cambios ayuda a seleccionar los casos de prueba para la prueba de regresión. d) No es correcto.

Analizar la trazabilidad entre la base de prueba, los objetos de prueba y los casos de prueba no ayuda a seleccionar datos de prueba para lograr la cobertura supuesta del objeto de prueba. La selección de datos de prueba está más relacionada con el análisis y la implementación de la prueba, no con la trazabilidad.

Pregunta #A6

FL-1.5.3 (K2)

Respuesta correcta: d.

- a) No es correcto. La calidad debe ser responsabilidad de todos los que trabajan en el proyecto y no es responsabilidad exclusiva del equipo de prueba.
b) No es correcto. En primer lugar, no es un beneficio si un equipo de pruebas externo no cumple con los plazos de entrega y, en segundo lugar, no hay razón para creer que los equipos de pruebas externos sentirán que no tienen que cumplir con plazos de entrega estrictos. c) No es correcto.

Es una mala práctica que el equipo de pruebas trabaje en completo aislamiento y esperaríamos que un equipo de pruebas externo se ocupara de cambiar los requisitos del proyecto y comunicarse bien con los desarrolladores. d) Es correcto. Las especificaciones nunca son perfectas, lo que significa que el desarrollador deberá hacer suposiciones. Un evaluador independiente es útil porque puede

Cuestionar y verificar las suposiciones y posterior interpretación realizadas por el desarrollador.

Pregunta #A7

FL-2.1.1 (K2)

Respuesta correcta: una

- a) Es correcto. En los modelos de desarrollo secuencial, en las fases iniciales, los evaluadores participan en revisiones de requisitos, análisis de pruebas y diseño de pruebas. El código ejecutable generalmente se crea en las fases posteriores, por lo que las pruebas dinámicas no se pueden realizar en las primeras etapas del SDLC. b) No es correcto. Las pruebas estáticas siempre se pueden realizar al principio del SDLC. c) No es correcto. La planificación de pruebas debe realizarse temprano en el SDLC antes de que comience el proyecto de prueba. d) No es correcto. Las pruebas de aceptación se pueden realizar cuando hay un producto en funcionamiento. En los modelos SDLC secuenciales, el producto funcional generalmente se entrega tarde en el SDLC.

Pregunta #A8

FL-2.1.4 (K2)

Respuesta correcta:c

- i. Es verdad. Un lanzamiento más rápido del producto y un tiempo de comercialización más rápido es una ventaja de DevOps. ii. Es falso. Normalmente, necesitamos menos esfuerzo para las pruebas manuales debido al uso de herramientas de prueba. automatización. III. Es verdad. La disponibilidad constante de software ejecutable es una ventaja. IV. Es falso. Se necesitan más pruebas de regresión. v. Es falso. No todo está automatizado y se debe configurar un marco de automatización de pruebas es caro. Por tanto c es correcto.

Pregunta #A9

FL-2.2.2 (K2)

Respuesta correcta:b

- a) No es correcto. El hecho de que el requisito sobre el rendimiento del sistema provenga directamente del cliente y que el rendimiento sea importante desde el punto de vista empresarial (es decir, alta prioridad) no hace que estas pruebas sean funcionales, porque no verifican "qué" hace el sistema. , sino "cómo" (es decir, qué tan rápido se procesan los pedidos). b) Es correcto. Este es un ejemplo de prueba de rendimiento, un tipo de prueba no funcional. pruebas. c) No es correcto. Según el escenario, no sabemos si interactuar con la interfaz de usuario es parte de las condiciones de prueba. Pero incluso si lo hiciéramos, el principal objetivo de estas pruebas es comprobar el rendimiento, no la usabilidad.

- d) No es correcto. No necesitamos conocer la estructura interna del código para realizar las pruebas de rendimiento. Se pueden ejecutar pruebas de eficiencia del desempeño sin conocimientos estructurales.

Pregunta #A10

FL-2.3.1 (K2)

Respuesta correcta: una

- a) Es correcto. Cuando se retira un sistema, esto puede requerir pruebas de migración de datos, que es una forma de prueba de mantenimiento.
- b) No es correcto. Las pruebas de regresión verifican si una solución afectó accidentalmente el comportamiento de otras partes del código, pero ahora estamos hablando de migración de datos a un nuevo sistema. c) No es correcto. Las pruebas de componentes se centran en componentes individuales de hardware o software, no en la migración de datos. d) No es correcto. Las pruebas de integración se centran en las interacciones entre componentes y/o sistemas, no en la migración de datos.

Pregunta #A11

FL-3.1.1 (K1)

Respuesta correcta: c

Solo no se puede revisar el código ejecutable de terceros. Por tanto la respuesta correcta es c.

Pregunta #A12

FL-3.1.3 (K2)

Respuesta correcta: d.

- i. Estos comportamientos son fácilmente detectables mientras el software se está ejecutando. Por eso, Se utilizarán pruebas dinámicas para identificarlos.
- ii. Este es un ejemplo de desviaciones de los estándares, que es un defecto típico que se más fácil de encontrar con pruebas estáticas. III. Si el software se ejecuta durante la prueba, se trata de una prueba dinámica.
- IV. Identificar defectos lo antes posible es el objetivo de las pruebas tanto estáticas como dinámicas.
- v. Este es un ejemplo de lagunas en la trazabilidad o cobertura de la base de la prueba, que es un defecto típico que se encuentra más fácilmente con las pruebas estáticas.
- Por tanto d es correcto.

Pregunta #A13

FL-3.2.2 (K2)

Respuesta correcta: b

- a) No es correcto. En todo tipo de revisiones hay más de un rol, incluso en los informales. b) Es correcto. Hay varias actividades durante el proceso de revisión formal.

- c) No es correcto. La documentación a revisar debe distribuirse lo antes posible.
posible.
- d) No es correcto. Se deben informar los defectos encontrados durante la revisión.

Pregunta #A14

FL-3.2.3 (K1)

Respuesta correcta: b

- a) No es correcto. Esta es la tarea del líder de revisión. b) Es correcto.
Esta es la tarea de la dirección en una revisión formal. c) No es correcto. Esta es la
tarea del moderador. d) No es correcto. Ésta es la tarea del
escriba.

Pregunta #A15

FL-4.2.2 (K3)

Respuesta correcta:

c Hay tres particiones de equivalencia: {..., 10, 11}, {12} y {13, 14, ...}. Los valores de frontera son 11,
12 y 13. En el análisis del valor de frontera de tres puntos para cada frontera, necesitamos probar la
frontera y sus dos vecinos, entonces:

- Para 11, probamos 10, 11, 12. •
- Para 12, probamos 11, 12, 13. • Para
- 13, probamos 12, 13, 14.

En total, necesitamos probar 10, 11, 12, 13 y 14.

- a) No es correcto. b)
No es correcto. c) Es
correcto. d) No
es correcto.

Pregunta #A16

FL-4.3.2 (K2)

Respuesta correcta: d.

- a) No es correcto. En este caso, todavía se necesita un caso de prueba ya que hay al menos una rama
(incondicional) por cubrir. b) No es correcto.
Cubrir sólo ramas incondicionales no implica cubrir todas las ramas condicionales.
- c) No es correcto. El 100% de cobertura de sucursal implica el 100% de cobertura del estado de cuenta,
no de otro modo. Por ejemplo, para una decisión IF sin ELSE, una prueba es suficiente para lograr
una cobertura de declaración del 100%, pero solo logra una cobertura de sucursal del 50%. d) Es
correcto. Cada resultado de decisión corresponde a una rama condicional, por lo que una cobertura de
rama del 100 % implica una cobertura de decisión del 100 %.

Pregunta #A17

FL-4.4.3 (K2)

Respuesta correcta:c

- a) No es correcto. El libro proporciona orientación general y no es un documento de requisitos formales, una especificación o un conjunto de casos de uso, historias de usuarios o negocios. procesos.
- b) No es correcto. Si bien se podría considerar la lista como un conjunto de cartas de prueba, se parece más a la lista de condiciones de prueba que se deben verificar. c) Es correcto. La lista de mejores prácticas de interfaz de usuario es la lista de condiciones de prueba para ser controlado sistemáticamente.
- d) No es correcto. Las pruebas no se centran en fallos que puedan ocurrir sino en el conocimiento de lo que es importante para el usuario, en términos de usabilidad.

Pregunta #A18

FL-4.5.1 (K2)

Respuesta correcta:b

- a) No es correcto. La redacción colaborativa de historias de usuario significa que todas las partes interesadas crean las historias de usuario de forma colaborativa para obtener una visión compartida.
- b) Es correcto. La escritura colaborativa de historias de usuario significa que todas las partes interesadas crean el historias de usuarios de forma colaborativa, para obtener la visión compartida.
- c) No es correcto. La redacción colaborativa de historias de usuario significa que todas las partes interesadas crean las historias de usuario de forma colaborativa para obtener una visión compartida.
- d) No es correcto. Esta es la lista de propiedades que debe tener cada historia de usuario, no la descripción del enfoque basado en colaboración.

Pregunta #A19

FL-5.1.1 (K2)

Respuesta correcta: d.

- a) No es correcto. El párrafo contiene información sobre los niveles de prueba y los criterios de salida, que forman parte del enfoque de prueba. b) No es correcto. El párrafo contiene información sobre los niveles de prueba y los criterios de salida, que forman parte del enfoque de prueba. c) No es correcto. El párrafo contiene información sobre los niveles de prueba y los criterios de salida, que forman parte del enfoque de prueba. d) Es correcto. El párrafo contiene información sobre los niveles de prueba y los criterios de salida, que forman parte del enfoque de prueba.

Pregunta #A20

FL-5.1.4 (K3)

Respuesta correcta:b

- a) No es correcto. Esta debe ser una actividad de equipo y no estar anulada por un solo equipo. miembro.
- b) Es correcto. Si las estimaciones de las pruebas no son las mismas pero la variación en los resultados es pequeña, se pueden aplicar reglas como "aceptar el número con más votos".

c) No es correcto. Aún no hay consenso ya que algunos dicen 13 y otros dicen 8. d) No es correcto. Una función no debe eliminarse sólo porque el equipo no puede

ponerse de acuerdo sobre las estimaciones de la prueba.

Pregunta #A21

FL-5.1.6 (K1)

Respuesta correcta:b

a) No es correcto. La pirámide de pruebas enfatiza tener una mayor cantidad de pruebas en los niveles de prueba más bajos. b) Es

correcto. No es cierto que cerca de la cima de la pirámide, la automatización de pruebas deba estar Más formal.

c) No es correcto. Por lo general, las pruebas de componentes y las pruebas de integración de componentes se automatizan mediante herramientas basadas

en API. d) No es correcto. Para las pruebas del sistema y las pruebas de aceptación, las pruebas automatizadas son normalmente se crea utilizando herramientas basadas en GUI.

Pregunta #A22

FL-5.2.1 (K1)

Respuesta correcta:c

a) No es correcto. El impacto del riesgo y la probabilidad del riesgo son independientes. b)

No es correcto. El impacto del riesgo y la probabilidad del riesgo son independientes. c) Es

correcto. El impacto del riesgo y la probabilidad del riesgo son independientes. d) No

es correcto. Necesitamos ambos factores para calcular el nivel de riesgo.

Pregunta #A23

FL-5.2.2 (K2)

Respuesta correcta: una

i. Riesgo del proyecto.

ii. Riesgo del producto.

III. Riesgo del producto.

IV. Riesgo del proyecto.

v. Riesgo del producto.

Por tanto a es correcto.

Pregunta #A24

FL-5.2.3 (K2)

Respuesta correcta: d.

a) No es correcto. Este es un ejemplo de una actividad de monitoreo de riesgos, no de análisis de riesgos. b) No es correcto. Este es un ejemplo de una decisión arquitectónica, no relacionada con pruebas.

c) No es correcto. Este es un ejemplo de cómo realizar un análisis de riesgo cuantitativo y no está relacionado con la minuciosidad o el alcance de las pruebas. d) Es correcto. Esto muestra cómo el análisis de riesgos afecta la minuciosidad de las pruebas (es decir, el nivel de detalle).

Pregunta #A25

FL-5.3.1 (K1)

Respuesta correcta: a, d

- a) Es correcto. El número de defectos encontrados está relacionado con la calidad del objeto de prueba.
- b) No es correcto. Esta es la medida de la eficiencia de la prueba, no la calidad del objeto de prueba. c) No es correcto. El número de casos de prueba ejecutados no nos dice nada sobre la calidad; los resultados de las pruebas podrían ser suficientes.
- d) Es correcto. La densidad de defectos está relacionada con la calidad del objeto de prueba. e) No es correcto. El tiempo de reparación es una métrica del proceso. No nos dice nada sobre la calidad del producto.

Pregunta #A26

FL-5.3.2 (K2)

Respuesta correcta: b

- a) No es correcto. Los impedimentos para las pruebas pueden ser de alto nivel y estar relacionados con el negocio, por lo que esta es una información importante para las partes interesadas del negocio.
- b) Es correcto. Las pruebas de rama son una métrica técnica utilizada por desarrolladores y evaluadores técnicos. Esta información no es de interés para los representantes comerciales.
- c) No es correcto. El progreso de la prueba está relacionado con el proyecto, por lo que puede resultar útil para los representantes comerciales.
- d) No es correcto. Los riesgos afectan la calidad del producto, por lo que puede resultar útil para los representantes comerciales.

Referencias

1. ISO/IEC/IEEE 29119-1 - Ingeniería de software y sistemas - Pruebas de software - Parte 1: Conceptos y definiciones, 2022.
2. ISO/IEC/IEEE 29119-2 - Ingeniería de software y sistemas - Pruebas de software - Parte 2: Prueba procesos, 2021.
3. ISO/IEC/IEEE 29119-3 - Ingeniería de software y sistemas - Pruebas de software - Parte 3: Documentación de prueba, 2013.
4. ISO/IEC/IEEE 29119-4 - Ingeniería de software y sistemas - Pruebas de software - Parte 4: Prueba técnicas, 2021.
5. ISO/IEC 25010 - Ingeniería de sistemas y software - Requisitos de calidad de sistemas y software-mentos y evaluación (SQuaRE) - Modelos de calidad de sistemas y software, 2011.
6. ISO/IEC 20246 - Ingeniería de software y sistemas - Revisiones de productos de trabajo, 2017.
7. ISO 31000 - Gestión de Riesgos, 2018.
8. "Probador certificado ISTQB - Programa de estudios de nivel básico v4.0", 2023.
9. "Estructura y reglas del examen ISTQB", 2021.
10. G. Myers, El arte de las pruebas de software (John Wiley and Sons, 2011)
11. A. Roman, Pruebas basadas en el pensamiento. El enfoque más razonable para el control de calidad, Naturaleza Springer, 2018.
12. J. Buxton y B. Randell, Redaktorzy Técnicas de ingeniería de software. Informe sobre una conferencia patrocinado por el Comité Científico de la OTAN, p. 16, 1969.
13. Z. Manna i R. Waldinger, "La lógica de la programación informática", IEEE Transactions on Ingeniería de software, volumen 4, n.º 3, págs. 199-229, 1978.
14. B. Boehm, Economía de la ingeniería de software (Prentice Hall, 1981)
15. A. Enders, "An Analysis of Errors and Their Causes in System Programs", IEEE Transactions on Software Engineering, tom 1, nr 2, págs. 140-149, 1975.
16. B. Beizer, Técnicas de prueba de software, Van Nostrand Reinhold, 1990.
17. C. Kaner, J. Bach y B. Pettichord, Lecciones aprendidas en pruebas de software: una experiencia basada en el contexto Enfoque, Wiley, 2011.
18. "UML 2.5 - Manual de referencia del lenguaje de modelado unificado", 2017. [En línea]. Disponible: [www. https://omg.org/spec/UML/2.5.1](http://www.omg.org/spec/UML/2.5.1).
19. "Plan de prueba de aceptación", [en línea]. Disponible: <https://ssdip.bip.gov.pl/objects/download/13576/zalacznik-nr-1-do-opz-szablon-pta-pdf.html> .
20. V. Stray, R. Florea i L. Paruch, "Exploración de los factores humanos del probador de software ágil", Software Quality Journal, tom 30, n.º 1, págs. 1-27, 2021.

21. L. Crispin y J. Gregory, Pruebas ágiles: una guía práctica para evaluadores y equipos ágiles, Pearson Educación, 2008.
22. R. Pressman, Ingeniería de software. El enfoque de un profesional, McGraw Hill, 2019.
23. T. Linz, Pruebas en Scrum: una guía para el aseguramiento de la calidad del software en el mundo ágil (Rocky Rincón, 2014)
24. G. Adzic, Especificación con el ejemplo: cómo los equipos exitosos entregan el software adecuado (Manning Publications, 2011)
25. D. ea Chelmsky, El libro de Rspec: Desarrollo impulsado por el comportamiento con Rspec, Cucumber y Friends, The Pragmatic Bookshelf, 2010.
26. M. Gärtner, ATDD con el ejemplo: una guía práctica para el desarrollo basado en pruebas de aceptación (Pearson Education, 2011)
27. G. Kim, J. Humble, P. Debois y J. Willis, El manual de DevOps, IT Revolution Press, 2016.
28. "Probador certificado ISTQB - Programa de estudios de nivel avanzado - Analista de pruebas", 2021.
29. "Probador certificado ISTQB - Programa de estudios de nivel avanzado - Analista de pruebas técnicas", 2021.
30. "Probador certificado ISTQB - Programa de estudios de nivel avanzado - Probador de seguridad", 2016.
31. C. Jones y O. Bonsignour, La economía de la calidad del software, Addison-Wesley, 2012.
32. S. Reid, "Revisiones de software utilizando ISO/IEC 20246", 2018. [En línea]. Disponible: <http://www.stureid.info/wp-content/uploads/2018/01/Software-Reviews.pdf>.
33. S. Nazir, N. Fatima i S. Chuprat, "Beneficios de la revisión del código moderno: hallazgos primarios de una revisión sistemática de la literatura", w ICSIM '20: Actas de la 3.ª Conferencia Internacional sobre Ingeniería de Software y Gestión de la Información, 2020.
34. T. Gilb y D. Graham, Inspección de software, Addison-Wesley, 1993.
35. M. Fagan, "Diseño e inspección de códigos para reducir errores en el desarrollo de programas", tom 15, n.º 3, págs. 182-211, 1975.
36. P. Johnson, Introducción a las revisiones técnicas formales (University College of London Press, 1996)
37. D. O'Neill, „Experimento nacional de calidad del software. Una lección de medición 1992-1997, 23º Taller Anual de Ingeniería de Software", Centro de Vuelo Espacial Goddard de la NASA, 1998.
38. K. Wiegers, Revisiones por pares en software: una guía práctica (Addison-Wesley Professional, 2001)
39. E. contra Veenendaal, The Testing Practitioner, UTN Publishers, 2004.
40. ISO, ISO 26262 - Vehículos de carretera - Seguridad funcional, 2011.
41. C. Sauer, "La eficacia de las revisiones técnicas del desarrollo de software: un programa de investigación motivado por el comportamiento", IEEE Transactions on Software Engineering, tom 26, nr 1, 2000.
42. "Glosario de términos de prueba de ISTQB", [en línea]. Disponible: <http://glossary/istqb.org>.
43. R. Craig y S. Jaskiel, Testomg de software sistemático, Artech House, 2002.
44. L. Copeland, Guía para profesionales sobre diseño de pruebas de software (Artech House, 2004)
45. T. Koomen, L. van der Aalst, B. Broekman i M. Vroon, TMap Next para pruebas basadas en resultados, Editores UTN, 2006.
46. P. Jorgensen, Pruebas de software, un enfoque artesanal, CRC Press, 2014.
47. P. Ammann i J. Offutt, Introducción a las pruebas de software, Cambridge University Press, 2016.
48. I. Forgács i A. Kovács, Diseño de pruebas prácticas: selección de diseños de pruebas tradicionales y automatizados técnicas, BCS, The Chartered Institute for IT, 2019.
49. G. O'Regan, Guía concisa para pruebas de software (Springer Nature, 2019)
50. A. Watson, D. Wallace i T. McCabe, "Pruebas estructuradas: una metodología de prueba utilizando la métrica de complejidad ciclomática", Departamento de Comercio de EE. UU., Administración de Tecnología, NIST, 1996.
51. B. Hetzel, La guía completa para pruebas de software (John Wiley and Sons, 1998)
52. A. Whittaker, Cómo descifrar software (Pearson, 2002)
53. J. Whittaker y H. Thompson, Cómo romper la seguridad del software, Addison-Wesley, 2003.
54. M. Andrews i J. Whittaker, Cómo romper el software web: pruebas funcionales y de seguridad de Aplicaciones web y servicios web, Addison-Wesley Professional, 2006.

55. J. Whittaker, Pruebas exploratorias de software. Consejos, Trucos, Recorridos y Técnicas para Guiar el Examen Diseño, Addison-Wesley, 2009.
56. C. Kaner, J. Falk y H. Nguyen, Pruebas de software informático, Wiley, 1999.
57. E. Hendrickson, ¡Explore! Reduzca el riesgo y aumente la confianza con pruebas exploratorias. El programador pragmático (2013)
58. B. Brykczynski, "Una encuesta de listas de verificación de inspección de software", ACM SIGSOFT Software Notas de ingeniería, tom 24, n.º 1, págs. 82-89, 1999.
59. J. Nielsen, "Mejora del poder explicativo de las heurísticas de usabilidad", en Actas de la Conferencia SIGCHI sobre factores humanos en sistemas informáticos: celebración de la interdependencia, 1994.
60. A. Gawande, El manifiesto de la lista de verificación: cómo hacer las cosas bien (Metropolitan Books, 2009)
61. M. Cohn, Historias de usuarios aplicadas al desarrollo de software ágil (Addison-Wesley, 2004)
62. B. Wake, "INVERTIR en buenas historias y tareas INTELIGENTES", 2003. [En línea]. Disponible: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.
63. G. Adzic, Cerrar la brecha de comunicación: especificación con ejemplos y pruebas de aceptación ágiles (Neuri Limited, 2009)
64. D. Jackson, M. Thomas i L. Millett, Redaktorzy, Software para sistemas confiables: ¿evidencia suficiente? Comité de Sistemas de Software Certificablemente Confiables, Consejo Nacional de Investigación, 2007.
65. S. Kan, Métricas y modelos en ingeniería de calidad de software (Addison-Wesley, 2003)
66. L. Westfall, Manual del ingeniero de calidad de software certificado (ASQ Quality Press, 2009)
67. M. Cohn, Tener éxito con Agile: desarrollo de software utilizando Scrum (Addison-Wesley, 2009)
68. B. Marick, "Exploración a través del ejemplo", 2003. [En línea]. Disponible: <http://www.exampler.es/old-blog/2003/08/21.1.html#agile-testing-project-1>.
69. K. Schwaber y M. Beedle, Desarrollo ágil de software con Scrum, Prentice-Hall, 2002.
70. R. Neeham, "Experiencia operativa con el sistema de acceso múltiple de Cambridge", w Computer Ciencia y Tecnología, Publicación de la conferencia, 1969.
71. Pandian, CR (2007) Gestión de Riesgos de Software Aplicada. Una guía para proyectos de software Gerentes, Publicaciones Auerbach, Boca Raton
72. Van Veenendaal, E (ed.) (2012) Pruebas prácticas basadas en riesgos, El enfoque PRISMA, Editores UTN: Países Bajos

Índice

A

Criterios de aceptación (CA), 56, 232
Desarrollo basado en pruebas de aceptación (ATDD), 77, 92

Pruebas de aceptación, 110
Revisión ad hoc, 162
Cobertura de todos los estados, 203
Cobertura de todas las transiciones, 204
Prueba alfa, 112
anomalía, 146
Autor (reseñas), 150

B

Desarrollo impulsado por el comportamiento (BDD), 77, 90
Pruebas beta, 112
Pruebas de caja negra, 122
Técnica de prueba de caja negra, 51, 172
Curva de Boehm, 43
Análisis de valor límite, 187
Sucursal, 214
Cobertura de sucursales, 214
Pruebas de sucursales, 214–217
Cheque de amigos, 154
Error, ver defecto
Gráfico de quemado, 291

C

Solicitud de cambio, 60
Lista de verificación, 226
Revisión basada en listas de verificación, 162

Pruebas basadas en listas de verificación, 226
Enfoque de prueba basado en la colaboración, 229
Pruebas de integración de componentes, 103
Rama condicional, 214
Gestión de configuración, 292
Sesgo de confirmación, 64
Pruebas de confirmación, 124
Entrega continua, 94
Despliegue continuo, 94
Integración continua, 94
Pruebas de aceptación contractual, 111
Directiva de control, 55
Cobertura, 47, 60
Priorización basada en cobertura, 270
Artículo de cobertura, 51, 57, 181, 193, 198, 203, 214

D

Tablero, 291
Depuración, 29
Minimización de la tabla de decisiones, 198
Prueba de tabla de decisiones, 194
Defecto, 29, 36
Gestión de defectos, 295
Enmascaramiento de defectos, 183
Informe de defecto, 56, 59, 295
Agrupación de defectos, 44
DevOps, 92–96
Canalización de DevOps, 94
Diseño impulsado por dominio, 77
Conductor, 52, 58, 100
Pruebas dinámicas, 135

mi

- Cobertura de cada elección, 184
- Pruebas tempranas, 43
- Criterios de entrada, 54, 258
- Partición de equivalencia (EP), 177
- errores, 36
- Error al adivinar, 220
- Hipótesis de error, 171
- Estimando, 260
- Estimación basada en ratios, 261
- Pruebas exhaustivas, 42
- Criterios de salida, 54, 258
- Técnica de prueba basada en la experiencia, 51, 173, 219.
- Pruebas exploratorias, 223
- Extrapolación, 262
- Programación extrema, 77, 88

F

- Facilitador, 150
- Fracaso, 29, 36, 39
- Falso negativo, 39
- Falso positivo, 39
- Fallo, ver defecto
- Desarrollo impulsado por funciones (FDD), 77
- Revisión formal, 145
- Tabla de transición completa, 202
- Requisito funcional, 115
- Pruebas funcionales, 115

Diagrama de Gantt, 268

- Condición de guardia, 200

I

- Análisis de impacto, 129
- Modelo incremental, 77
- Revisión individual, 146
- Revisión informal, 154
- Inspección, 155
- Estrategia de integración, 105
- Pruebas de integración, 103
- Planificación de iteraciones, 257
- Modelo iterativo, 77, 78.

k

- Kanban, 77, 84

I

- TI ajustada, 77
- Pruebas de aceptación de cumplimiento legal, 112

METRO

- Pruebas de mantenimiento, 127
- Gerente (opiniones), 150
- Métrica, 287
- Error, ver Error
- Objeto simulado, 52, 58
- Moderador, ver Facilitador

NOTA

- Pruebas no funcionales, 117
- Cobertura del conmutador N, 204

oh

- Pruebas de aceptación operativa, 111

PIAO

- Revisión de pareja, 154
- Regla de Pareto, 44
- Revisión por pares, 152
- Lectura basada en perspectiva, 163
- Planificación del póquer, 262
- Riesgo del producto, 280
- Riesgo del proyecto, 279
- Modelo de prototipo, 77, 82

q

- Calidad, 34
- Garantía de calidad (QA), 36
- Control de calidad (QC), 36

R

- Registrador, ver Escribano
- Pruebas de regresión, 125
- Planificación de lanzamiento, 257
- Requisitos, 50
- Priorización basada en requisitos, 270
- retrospectiva, 97
- Revisión, 143
- Revisor, 151
- Líder de revisión, 151

Reunión de revisión, 149	nivel alto, 57 nivel
Proceso de revisión, 145	bajo, 58
Riesgo, 55, 279	Finalización de la prueba, 52
Análisis de riesgos, 281	Informe de finalización de la prueba, 60, 288
Evaluación de riesgos, 282	Condición de prueba, 50, 56
Priorización basada en riesgos, 270	Control de prueba, 48, 286
Pruebas basadas en riesgos, 278	Datos de prueba, 51, 57, 58
Control de riesgos, 284	Diseño de pruebas, 51
Identificación de riesgos, 281	Desarrollo basado en pruebas (TDD), 77, 89
Impacto del riesgo, 279	Esfuerzo de prueba, 259
Nivel de riesgo, 279	Entorno de prueba, 52, 57, 58
Probabilidad de riesgo, 279	Probador, 63
Gestión de riesgos, 277	Ejecución de prueba, 52
Matriz de riesgos, 282	Calendario de ejecución de pruebas, 58, 268
Mitigación de riesgos, 284	Prueba primero, 88
Monitoreo de riesgos, 286	Arnés de prueba, 100
Registro de riesgos, 54	Implementación de pruebas, 51
Revisión basada en roles, 163	Pruebas, 27
Causa raíz, 40	Prueba de independencia, 67
	Principios de prueba, 41
S	Cuadrantes de prueba, 276
Escenarios y simulacros, 163	Nivel de prueba, 99, 122
Escriba, 150	Registro de prueba, 59
Melé, 77, 83	Responsable de pruebas, 62
Modelo secuencial, 77	Monitoreo de pruebas, 48, 286
Virtualización de servicios, 52, 58	Objeto de prueba, 27, 99
Mayús-izquierda, 43, 96	Objetivo de la prueba, 28
Simulador, 52, 58	Plan de prueba, 48, 54, 253
Habilidades, 35, 64	Planificación de pruebas, 48, 253
Ciclo de vida de desarrollo de software (SDLC), 76, 87	Procedimiento de prueba, 51, 58
Modelo espiral, 77, 81	Proceso de prueba, 47, 53
Cobertura de declaración, 214	Informe de progreso de la prueba, 55, 288
Prueba de declaración, 212	Pirámide de prueba, 275
Tabla de estados, 202	Resultado de la prueba, 39
Diagrama de transición de estado, 200, 202	Guión de prueba, 58
Pruebas de transición estatal, 199	Conjunto de prueba, 58
Análisis estático, 134	Estrategia de prueba, 255
Pruebas estáticas, 134	Técnica de prueba, 171
Cobertura estructural, 120	Tipo de prueba, 99, 122
Talón, 52, 58, 100	Software de prueba, 54
Subsunción, 218	Estimación de tres puntos, 265 BVA de 3
Pruebas de integración de sistemas, 103	valores, 189
Pruebas del sistema, 107	Herramientas, 307
	Trazabilidad, 60 BVA
	de 2 valores, 189
t	
Tablero de tareas, 291	Ud.
Velocidad del equipo, 257	Rama incondicional, 214
Revisión técnica, 154	Proceso unificado (UP), 77, 81
Análisis de pruebas, 49	Caso de uso, 208
Enfoque de prueba, 253	Pruebas de casos de uso, 208
Automatización de pruebas, 307	Pruebas de aceptación de usuario (UAT), 111
Base de prueba, 49	Historia de usuario (EE. UU.), 229
Caso de prueba (TC), 51, 57, 59	Puntos de historia de usuario, 264

V

- Validación, 27, 46
- Cobertura de transiciones válidas, 203
- Verificación, 27, 46
- modelo V, 77, 80

- Técnica de prueba de caja blanca, 51, 172, 211–219
- Enfoque de equipo completo, 66
- Delphi de banda ancha, 262
- Estructura de desglose del trabajo (EDT), 260
- Producto del trabajo, 54, 135

W.

- Tutorial, 154
- Modelo cascada, 77, 79
- Pruebas de caja blanca, 120

- Cobertura del interruptor Z 0, 203