Preguntas de trivia

Primera sección – Técnicas de pruebas

a) Clases equivalentes
b) Análisis de valores de frontera
c) Tablas de decisión
d) Pairwise testing
e) Diagramas de transición de estados
f) Cobertura de sentencias
g) Cobertura de decisiones
h) Predicción de error
i) Pruebas exploratorias
j) Pruebas basadas en riesgos

Es una técnica de pruebas que sirve para probar software con una lógica de negocio compleja, o que depende de varias condiciones. Se expresa en dos partes: condiciones y acciones	C
Es una técnica de caja blanca que analiza la cobertura del código, midiendo qué porcentaje de las líneas de código son ejecutadas al menos una vez durante las pruebas	F
Es una técnica de prueba basada en la experiencia, que consiste en un acercamiento para aprender del componente o sistema, y diseñar pruebas donde más se necesite	I
Es una técnica de pruebas que se apoya en un gráfico que representa estados, transiciones y acciones	E
Tratan de anticipar dónde se podría presentar un error en el software, tomando en cuenta aspectos como la experiencia del tester, fallos comunes de ese tipo de software o del equipo desarrollador.	H
Es una técnica de pruebas que se basa en la teoría de conjuntos, que dice que “los objetos de la misma clase se parecen entre sí”	A
Esta técnica, ofrece un método para establecer una cobertura del 97% de todos los posibles casos, pero con muchos menos casos de prueba	D
Es una técnica que extiende la técnica de clases equivalentes, al exigir probar los límites de las clases o particiones que se hayan definido	B
Se trata de una estrategia para evaluar en qué partes del sistema enfocar las pruebas, por ser las que mayor riesgo representan.	J
Es una técnica de caja blanca que explora todas las ramas o decisiones que ocurren al correr el código, también se le conoce como pruebas de camino básico	G

Segunda sección – Pruebas automatizadas

a) Pruebas basadas en ejemplos
b) Pruebas basadas en propiedades
c) Capture/Playback
d) Data driven
e) BDD
f) Simple fuzzing
g) Round trip
h) Invariantes
i) Idempotentes
j) Test Oracle

Es un framework de automatización que consiste en capturar lo que sucede en la pantalla (clics, teclazos y movimientos del puntero del mouse), y posteriormente reproducir esos movimientos	C
Se trata de una estrategia de desarrollo y pruebas de software que evolucionó desde el Test Driven Development, y usa un lenguaje llamado Gherkin	E
Propiedades o elementos que no deben ser modificados, aunque se les aplique una función	H
Principio que indica que si al resultado de una función, le aplicamos la función inversa, el resultado debería ser idéntico al original	G
Son muy útiles cuando se debe refactorizar software, o pasarlo a un nuevo lenguaje	J
Este tipo de pruebas automatizadas replican las pruebas manuales con mayor eficacia y velocidad	A
Se trata de efectos que una sola vez se pueden aplicar a elementos de software	I
Framework en que se define una fuente de datos externa y se usa como datos de prueba	D
Patrón para probar validaciones a través de datos inválidos aleatorios, tratando de hacer que el software falle	F
Tipo de pruebas automatizadas en donde se describen las entradas y los resultados, y automáticamente se ejecutan las pruebas muchas veces, comparando los resultados contra las propiedades que fueron descritas	B

Tercera sección – Gestión de pruebas

a) DevOps
b) CI/CD
c) CTP
d) STEP
e) TPI
f) TMM
g) TDD
h) BDD

Modelo que describe los procesos más importantes en las pruebas de software. Se suele aplicar en sistemas donde es crítico el funcionamiento correcto del software.	C
Proceso que incorpora la integración y la distribución del software de manera continua, y durante todo el ciclo de vida.	B
Estrategia de desarrollo que evolucionó del TDD, y está enfocada en mejorar la comunicación entre desarrolladores, gestores de proyectos y equipo de ventas.	H
Es un modelo usado para evaluar y mejorar la capacidad de los procesos de prueba de una organización.	F
Filosofía que combina desarrollo de software con sus operaciones de forma integrada.	A
Conjunto de prácticas y metodologías utilizadas para planificar, diseñar, ejecutar y analizar pruebas de software.	D
Metodología de desarrollo de software enfocada en escribir primero las pruebas, y después codificar el software capaz de pasar esas pruebas.	G
Metodología para mejorar y optimizar los procesos de prueba de software.	E

Relación

Aplicación de técnicas

Gestión de pruebas y miscelánea.

1 Pairwise testing	3 Para aplicar esta técnica, primero se deben entender los requerimientos, y dividirlos en condiciones y en acciones que el software realizará en consecuencia
2 Cobertura de decisiones	6 En esta técnica, se debe verificar que cada línea de código sea ejecutada al menos una vez
3 Tablas de decisión	5 Consiste en dividir los elementos en clases y probar algún elemento de cada una
4 Diagramas de transición de estados	2 Su métrica más común se calcula dividiendo el número de ramas probadas entre el número de ramas existentes
5 Clases equivalentes	1 Ante la imposibilidad de hacer pruebas exhaustivas, se suele requerir a esta técnica de pruebas
6 Cobertura de sentencias	4 Para aplicar esta técnica, se debe elaborar un diagrama donde se establezcan estados y transiciones. La cobertura de las pruebas debe ser por ambos.

1 Pruebas automatizadas	4 Técnica de evaluación de riesgos más usada
2 UX	5 Aspecto del software que se puede probar de manera automatizada con uno de los frameworks
3 DevOps	1 Sus principios son que deben ser automatizables, reutilizables, independientes, rápidas y ofrecer una total cobertura
4 Matriz de probabilidad e impacto	2 Por su naturaleza subjetiva, es una característica casi imposible de probar de manera automatizada
5 UI	3 Su objetivo es entregar de manera continua software de alta calidad y mejorar la eficiencia en el ciclo de vida, tanto en el desarrollo como en el despliegue
6 TMM	6 Toma de otro modelo aspectos como los niveles de madurez, o las áreas de proceso, así como prácticas.