Técnicas de pruebas de caja negra

*El objetivo de una técnica de prueba es ayudar a identificar las condiciones de pruebas, los casos de pruebas y los datos de pruebas*

Contenido

[Técnicas basadas en la especificación o técnicas de caja NEGRA 3](#_Toc80629428)

[**1.** Partición de equivalencia 3](#_Toc80629429)

[**2.** Análisis de Valores Frontera 4](#_Toc80629430)

[**3.** Pruebas de tabla de decisión 5](#_Toc80629431)

[**4.** Prueba de transición de estado 6](#_Toc80629432)

[**5.** Prueba de Caso de Uso 7](#_Toc80629433)

# Técnicas basadas en la especificación o técnicas de caja NEGRA

1. Partición de equivalencia

La partición de equivalencia divide los datos en particiones (también conocidas como clases de equivalencia) de tal manera que se espera que todos los miembros de una partición dada sean procesados de la misma manera. Existen particiones de equivalencia tanto para valores válidos como no válidos.

1. Los valores válidos son los valores que debe aceptar el componente o el sistema. Una partición de equivalencia que contiene valores válidos se llama “Partición de equivalencia válida”
2. Los valores no válidos son valores que deben ser rechazados por el componente o el sistema. Una partición de equivalencia que contiene valores no validos se llama “partición de equivalencia no valido”.
3. Las particiones pueden identificarse para cualquier elemento de dato relacionado con el objeto de prueba, incluyendo entradas, salidas valores internos, valores relacionados con el tiempo (ejemplo, antes o después de un evento) y para parámetro de interfaz (por ejemplo, componentes integrados que se están probando durante la prueba de integración).
4. Cualquier partición se puede dividir en subparticiones si fuera necesario.
5. Cada valor debe pertenecer a una y solo a una partición de equivalencia.

Cuando se usan las particiones de equivalencia no válidas en caso de pruebas, debe probarse individualmente, es decir, no combinarse con otras particiones de equivalencia no valida, para garantiza que no se produzca un enmascaramiento de los fallos. Los fallos se pueden enmascarar cuando se producen varios fallos al mismo tiempo, pero solo uno de ellos es visible, lo que hace que los otros fallos queden sin detectar.

Para lograr una cobertura del 100 % con esta técnica, los casos de pruebas deben cubrir todas las particiones identificadas (incluidas las particionas no válidas) utilizando, como mínimo, un valor de cada partición. La cobertura se mide como el numero de particiones de equivalencia probadas por al menos un valor, dividido por el número total de particiones de equivalencia identificadas, normalmente expresado como un porcentaje. La partición de equivalencia es aplicable en todos los niveles de pruebas.

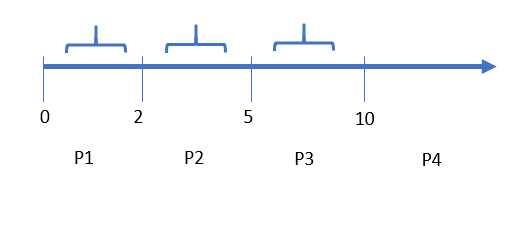
**Ejemplo:**

Pregunta 25

Se debe calcular la bonificación de un empleado. **No puede ser negativa**, pero se puede reducir a cero.

1. Menos de 2 o igual a 2 años
2. Más de 2 años, pero menos de 5 años
3. 5 a 10 años inclusive o más de 10 años

¿Cuál es el número mínimo de casos de pruebas necesario para cubrir todas las particiones de equivalencia válidas para calcular la bonificación?



Opciones:

1. 3
2. 5
3. 2
4. 4
5. Análisis de Valores Frontera

El análisis de valores de fronteras (AVF) es una extensión de la partición de equivalencia, pero solo se puede utilizar cuando la partición está ordenada, y consiste en datos numéricos o secuenciales. Los valores mínimo y máximo (o valores inicial y final) de una partición son sus valores de frontera.

Por ejemplo, suponer que un campo de entrada acepta un único valor entero como entrada, utilizando un teclado numérico para limitar las entradas de modo que las entradas no enteras sean imposibles. El rango valido es de 1 a 5, ambos valores incluidos. Por lo tanto, hay tres particiones de equivalencia: invalida (demasiada baja); valida; invalida (demasiada alta), los valores frontera son 6 y 9. Para la participación invalidad (demasiada baja), solo hay un valor de frontera, 0, porque se trata de una partición con un solo miembro.

En el ejemplo anterior, se identifican dos valores frontera por frontera. La frontera entre invalido (demasiado bajo) y valido proporciona los valores de prueba 0 y 1. La frontera entre válido en invalido (demasiado alto) proporciona los valores de prueba entre 5 y 6. Algunas variantes de esta técnica identifican tres valores frontera por frontera: los valores antes, en y justo después de la frontera. En el ejemplo anterior, utilizando tres puntos para los valores frontera, los valores de la prueba de frontera inferior son 0,1, y 2, y los valores de la prueba frontera superior son 4, 5, 6.

El comportamiento en las fronteras de las particiones de equivalencia y más probable que sea incorrecto que el comportamiento dentro de las peticiones. Es importante recordar que tanto las fronteras especificadas como las implementadas pueden desplazarse a posiciones por encima o por debajo de sus posiciones prevista, pueden omitirse por completo o pueden complementarse con fronteras adicionales no deseadas, El análisis y la prueba del valor frontera revelarán casi todos los defectos forzando al software a mostrar comportamientos de una partición distinta a la que debería pertenecer el valor frontero.

El análisis de valores frontera se puede aplicar en todos los niveles de prueba. Esta técnica se utiliza generalmente para probar los requisitos que requieren un rango de números (incluyendo fecha y horas). La cobertura de frontera para una partición se mide como el número de valores frontera probados, dividido por el número total de valores de frontera de prueba identificados, normalmente expresado como un porcentaje.

1. Pruebas de tabla de decisión

Las técnicas de pruebas combinatorias son útiles para probar la implementación de requisitos de sistema que especifican como diferentes combinaciones de condiciones generan diferentes resultados. Un enfoque para este tipo de prueba es la prueba de tabla de decisión.

Las tablas de decisión son una buena manera de documentar reglas de negocios complejas que un sistema debe implementar, Al crear tablas de decisión, el probador identifica las condiciones (A menudo entradas) y las acciones resultantes (a menudo salidas) de sistema. Estas conforman las filas de la tabla, generalmente con las condiciones en la parte superior y las acciones de la parte inferior. Cada columna corresponde a una regla de decisión que define una combinación única de condiciones que resulta en la ejecución de las acciones asociadas a esa regla, Los valores de las condiciones y acciones, normalmente, se muestran como valores booleanos (verdadero y falso) o valores discretos (por ejemplo, rojo, verde, azul), pero también pueden ser números o intervalos de números. Estos diferentes tipos de condiciones y acciones pueden estar juntos en la misma tabla.

La notación habitual en las tablas de decisión es la siguientes:

* S significa que la condición es verdadera
* N significa que la condición es falsa
* Un guion significa que el valor de la condición no importa.



Para las acciones

* X significa que la acción debe ocurrir
* En blanco significa que la acción no debe ocurrir

Una tabla de decisión completa tiene suficientes columnas para cubrir cada combinación de condiciones. La tabla se puede colapsar borrando las columnas que contiene combinaciones de condiciones imposibles, las columnas que contiene en combinaciones de condiciones posibles, pero no factibles y las columnas que prueban combinaciones de condiciones que no afecten el resultado.

La cobertura estándar mínima habitual para la prueba de tabla de decisión es tener al menos un caso de prueba por regla de decisión de la tabla. Esto implica, normalmente, cubrir todas las combinaciones de condiciones. La cobertura se mide como el número de reglas de decisión probadas por, al menos, un caso de prueba, dividido por el número total de reglas de decisión, normalmente expresado como un porcentaje.

La fortaleza de la prueba de tabla de decisión es que ayuda a identificar todas las combinaciones importantes de condiciones, algunas de las cuales, de otra manera, podrían ser ignoradas. También ayuda a encontrar cualquier desfase en los requisitos. Puede aplicarse a toda la situación en las que el comportamiento del software depende de una combinación de condiciones, en cualquier nivel de prueba.

1. Prueba de transición de estado

Los componentes o sistemas pueden responder de forma diferente a un evento dependiendo de las condiciones del momento o de su historia previa **(por ejemplo,** los eventos que ha ocurrido desde que se inicializo el sistema). La historia previa puede resumirse utilizando el concepto de estado. Un diagrama de transición de estado muestra los posibles estados del software, así como en la forma en que el software entra, sale y realiza la transición entre estados. Una transición se inicia con un evento (por ejemplo, la entrada de un valor por parte del usuario en un campo). El evento resulta en una transición. Si el mismo evento puede resultar en dos o mas transiciones diferentes desde el mismo estado, ese evento puede estar condicionado por una condición de guarda. El cambio de estado puede provocar que el software tome una acción (Por ejemplo, emitir el resultado de un cálculo o un mensaje de error).

Una tabla de transición de estado muestra todas las transiciones válidas y las transiciones potencialmente invalidas entre estados, así como los eventos, las condiciones de guarda y las acciones resultantes para las transiciones válidas. Los diagramas de transición de estados, normalmente, soló muestran las transiciones válidas y excluyen las transiciones no válidas.

Las pruebas pueden ser diseñadas para cubrir una secuencia de estados típica, para practicar todos los estados, para practicar cada transición, para practicar secuencias especificas de transiciones, o para probar transiciones inválidas.

La prueba de transición de estados se basa en aplicaciones basadas en menús y es extensamente utilizada en la industria de software embebido. La técnica también es adecuada para modelar un escenario de negocio con estados específicos o para probar la navegación en pantalla. El concepto de estado abstracto: puede representar unas pocas líneas de código o todo un proceso de negocio.

La cobertura se mide, habitualmente, como el numero de estados o transiciones identificados probados, dividido por el número total de estados o transiciones identificados en el objeto de prueba, normalmente expresado como un porcentaje.

1. Prueba de Caso de Uso

Las pruebas se pueden obtener a partir de casos de usos, que son una forma especifica de diseñar interacciones con elementos de software, incorporando requisitos para las funciones del software representadas por el caso de uso, Los casos de usos no están asociados con actores (Usuarios, humanos, hardware externo u otros componentes o sistema) y sujetos (el componente o sistema al que se le aplica el caso de uso).

Cada caso de uso especifica algún comportamiento que un sujeto puede realizar en colaboración con uno o mas actores. Un caso de uso puede describirse mediante interacciones y actividades, así como mediante precondiciones, post condiciones y lenguaje natural cuando resulte adecuado. Las interacciones entre los actores y el sujeto pueden resultar en cambios en el estado del sujeto. Las interacciones pueden representarse gráficamente mediante flujos de trabajo, diagramas de actividad o modelos de procesos de negocio.

Un caso de uso puede incluir posibles variaciones de su comportamiento básico, incluyendo el comportamiento excepcional y de errores (respuesta del sistema y recuperación de errores de programación, de aplicación y de comunicación, por ejemplo, resultando en un mensaje de error). La cobertura se puede medir por el porcentaje de comportamiento de casos de uso probados divido por el numero total de comportamientos de caso de uso, normalmente expresado como un porcentaje.