

El estándar internacional (ISO/IEC/IEEE 29119-4) contiene descripciones de técnicas de prueba y sus correspondientes medidas de cobertura (ver Craig 2002 y Copeland 2004 para más información sobre técnicas).

## 4.2 Técnicas de Prueba de Caja Negra

### 4.2.1 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 (ver Kaner 2013 y Jorgensen 2014). Existen particiones de equivalencia tanto para valores válidos como no válidos.

- 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".
- Los valores no válidos son valores que deben ser rechazados por el componente o sistema. Una partición de equivalencia que contiene valores no válidos se llama "partición de equivalencia no válida".
- Las particiones pueden identificarse para cualquier elemento de datos relacionado con el objeto de prueba, incluyendo entradas, salidas, valores internos, valores relacionados con el tiempo (por ejemplo, antes o después de un evento) y para parámetros de interfaz (por ejemplo, componentes integrados que se están probando durante la prueba de integración).
- Cualquier partición se puede dividir en subparticiones si fuera necesario.
- Cada valor debe pertenecer a una y sólo a una partición de equivalencia.
- Cuando se utilizan particiones de equivalencia no válidas en casos de prueba, deben probarse individualmente, es decir, no combinarse con otras particiones de equivalencia no válidas, para garantizar que no se produzca un enmascaramiento de los fallos. Los fallos se pueden enmascarar cuando se producen varios fallos al mismo tiempo, pero sólo 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 prueba deben cubrir todas las particiones identificadas (incluidas las particiones no válidas) utilizando, como mínimo, un valor de cada partición. La cobertura se mide como el número 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 a todos los niveles de prueba.

### 4.2.2 Análisis de Valores Frontera

El análisis de valores frontera (AVF) es una extensión de la partición de equivalencia, pero sólo 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 frontera (Beizer 1990).

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 válido es de 1 a 5, ambos valores incluidos. Por lo tanto, hay tres particiones de equivalencia: inválida (demasiado baja); válida; inválida (demasiado alta). Para la partición de equivalencia válida, los valores frontera son 1 y 5. Para la partición no válida (demasiado alta), los valores frontera son 6 y 9. Para la partición inválida (demasiado baja), sólo hay un valor 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 inválido (demasiado bajo) y válido proporciona los valores de prueba 0 y 1. La frontera entre válido e inválido (demasiado alto) proporciona los valores de prueba 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 la frontera inferior son 0, 1 y 2, y los valores de la prueba de la frontera superior son 4, 5 y 6 (Jorgensen 2014).

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

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 fechas 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 frontera de prueba identificados, normalmente expresado como un porcentaje.

### 4.2.3 Prueba de Tabla de Decisión

Las técnicas de prueba combinatorias son útiles para probar la implementación de requisitos de sistema que especifican cómo 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 negocio 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) del sistema. Éstas conforman las filas de la tabla, generalmente con las condiciones en la parte superior y las acciones en 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 o 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 siguiente:

Para las condiciones:

- $S^{43}$  significa que la condición es verdadera (también se puede mostrar como  $V^{44}$  o 1).
- $N^{45}$  significa que la condición es falsa (también se puede mostrar como  $F^{46}$  o 0).
- Un guion "-" significa que el valor de la condición no importa (también puede mostrarse como  $N/A^{47}$ ).

<sup>43</sup> Consultar \*\*

<sup>44</sup> Consultar \*\*

<sup>45</sup> Consultar \*\*

<sup>46</sup> Consultar \*\*

<sup>47</sup> Consultar \*\*

Para las acciones:

- X significa que la acción debe ocurrir (también puede mostrarse como S o V o 1).
- En blanco significa que la acción no debe ocurrir (también puede mostrarse como - o N o F o 0).

Una tabla de decisión completa<sup>48</sup> tiene suficientes columnas para cubrir cada combinación de condiciones. La tabla se puede colapsar borrando las columnas que contienen combinaciones de condiciones imposibles, las columnas que contienen combinaciones de condiciones posibles pero no factibles y las columnas que prueban combinaciones de condiciones que no afectan al resultado. Para más información sobre cómo colapsar las tablas de decisión, consultar el Programa de Estudio de Nivel Avanzado de Analista de Pruebas del ISTQB (ISTQB-ATA).

La cobertura estándar mínima habitual para la prueba de tabla de decisión es tener al menos un caso de prueba por regla de decisión en 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<sup>49</sup> en los requisitos. Puede aplicarse a todas las situaciones en las que el comportamiento del software depende de una combinación de condiciones, en cualquier nivel de prueba.

### 4.2.4 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 han ocurrido desde que se inicializó 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 la forma en que el software entra, sale y realiza las transiciones 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 más transiciones diferentes desde el mismo estado, ese evento puede estar condicionado por una condición de guarda<sup>50</sup>. 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 inválidas entre estados, así como los eventos, las condiciones de guarda y las acciones resultantes para las transiciones válidas. Los diagramas de transición de estado, normalmente, sólo 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 específicas de transiciones, o para probar transiciones inválidas.

La prueba de transición de estado se utiliza para aplicaciones basadas en menús y es extensamente utilizada en la industria del 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 es abstracto: puede representar unas pocas líneas de código o todo un proceso de negocio.

<sup>48</sup> Consultar \*\*

<sup>49</sup> Consultar \*\*

<sup>50</sup> Consultar \*\*

La cobertura se mide, habitualmente, como el número 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. Para más información sobre los criterios de cobertura para la prueba de transición de estado, consultar el Programa de Estudio de Nivel Avanzado de Analista de Pruebas del ISTQB (ISTQB-ATA).

### 4.2.5 Prueba de Caso de Uso

Las pruebas se pueden obtener<sup>51</sup> a partir de casos de uso, que son una forma específica de diseñar interacciones con elementos software, incorporando requisitos para las funciones del software representadas por los casos de uso. Los casos de uso están asociados con actores (usuarios humanos, hardware externo u otros componentes o sistemas) y sujetos (el componente o sistema al que se aplica el caso de uso).

Cada caso de uso especifica algún comportamiento que un sujeto puede realizar en colaboración con uno o más actores (UML 2.5.1 2017). Un caso de uso puede describirse mediante interacciones y actividades, así como mediante precondiciones, poscondiciones 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 tratamiento de un 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). Las pruebas están diseñadas para practicar las conductas definidas (básicas, excepcionales o alternativas, y tratamiento de errores). La cobertura se puede medir por el porcentaje de comportamientos de casos de uso probados dividido por el número total de comportamientos de casos de uso, normalmente expresado como un porcentaje.

Para obtener más información sobre los criterios de cobertura para la prueba de caso de uso, consultar el Programa de Estudio de Nivel Avanzado de Analista de Pruebas del ISTQB (ISTQB-ATA).

## 4.3 Técnicas de Prueba de Caja Blanca

Las técnicas de prueba de caja blanca se basan en la estructura interna del objeto de prueba. Las técnicas de prueba de caja blanca se pueden utilizar en todos los niveles de prueba, pero las dos técnicas relacionadas con el código que se discuten en esta sección se utilizan con mayor frecuencia en el nivel de prueba de componente. Hay técnicas más avanzadas que se utilizan en algunos entornos de seguridad crítica, de misión crítica o de alta integridad para lograr una cobertura más completa, pero no se tratan en este documento. Para más información sobre estas técnicas, consultar el Programa de Estudio de Analista de Pruebas Técnicas de Nivel Avanzado del ISTQB.

### 4.3.1 Prueba y Cobertura de Sentencia

La prueba de sentencia practica las sentencias ejecutables en el código. La cobertura se mide como el número de sentencias ejecutadas por las pruebas dividido por el número total de sentencias ejecutables en el objeto de prueba, normalmente expresado como un porcentaje.

<sup>51</sup> Consultar \*\*