

2.3.3 Prueba de Caja Blanca

La prueba de caja blanca obtiene pruebas basadas en la estructura interna del sistema o en su implementación. La estructura interna puede incluir código, arquitectura, flujos de trabajo y/o flujos de datos dentro del sistema (ver sección 4.3).

Se puede medir la intensidad de la prueba de caja blanca a través de la cobertura estructural. La cobertura estructural es la medida en que algún tipo de elemento estructural ha sido practicado mediante pruebas, y se expresa como un porcentaje del tipo de elemento cubierto.

En el nivel de prueba de componente, la cobertura de código se basa en el porcentaje de código del componente que ha sido probado, y puede medirse en términos de diferentes aspectos del código (elementos de cobertura), tales como el porcentaje de sentencias ejecutables probadas en el componente, o el porcentaje de resultados de decisión probados. Estos tipos de cobertura se denominan, de forma colectiva, cobertura de código. En el nivel de prueba de integración de componentes, la prueba de caja blanca pueden basarse en la arquitectura del sistema, como las interfaces entre componentes, y la cobertura estructural puede medirse en términos del porcentaje de interfaces practicadas por las pruebas.

El diseño y la ejecución de la prueba de caja blanca pueden implicar competencias o conocimientos especiales, como la forma en que se construye el código (por ejemplo, para utilizar herramientas de cobertura de código), cómo se almacenan los datos (por ejemplo, para evaluar posibles consultas a la base de datos), y cómo utilizar las herramientas de cobertura e interpretar correctamente sus resultados.

2.3.4 Prueba Asociada al Cambio

Cuando se realizan cambios en un sistema, ya sea para corregir un defecto o debido a una funcionalidad nueva o modificada, se debe probar para confirmar que los cambios han corregido el defecto o implementado la funcionalidad correctamente, y no han causado ninguna consecuencia adversa imprevista.

- Prueba de confirmación: Una vez corregido un defecto, el software se puede probar con todos los casos de prueba que fallaron debido al defecto, que se deben volver a ejecutar en la nueva versión de software. El software también puede probarse con nuevas pruebas si, por ejemplo, el defecto consistía en la falta de una funcionalidad. Como mínimo, los pasos para reproducir el fallo o los fallos causados por el defecto deben volver a ejecutarse en la nueva versión del software. El objetivo de una prueba de confirmación es confirmar que el defecto original se ha solucionado de forma satisfactoria.
- Prueba de regresión: Es posible que un cambio hecho en una parte del código, ya sea una corrección u otro tipo de cambio, pueda afectar accidentalmente el comportamiento de otras partes del código, ya sea dentro del mismo componente, en otros componentes del mismo sistema, o incluso en otros sistemas. Los cambios pueden incluir modificaciones en el entorno, tales como una nueva versión de un sistema operativo o de un sistema de gestión de bases de datos. Estos efectos secundarios no deseados se denominan regresiones. La prueba de regresión implica la realización de pruebas para detectar estos efectos secundarios no deseados.

La prueba de confirmación y la prueba de regresión se realizan en todos los niveles de prueba.

Especialmente en los ciclos de vida de desarrollo iterativos e incrementales (por ejemplo, Agile), las nuevas características, los cambios en las características existentes y la refactorización del código dan como resultado cambios frecuentes en el código, lo que también requiere pruebas asociadas al cambio. Debido a la naturaleza evolutiva del sistema, la prueba de confirmación y la prueba de regresión son muy importantes. Esto es particularmente relevante para los sistemas de Internet de las Cosas, donde los objetos individuales (por ejemplo, los dispositivos) se actualizan o reemplazan con frecuencia.

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 obtener⁵¹ 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.

⁵¹ Consultar **

4.3.2 Prueba y Cobertura de Decisión

La prueba de decisión practica las decisiones en el código y prueba el código que se ejecuta basado en los resultados de la decisión. Para ello, los casos de prueba siguen los flujos de control que se producen desde un punto de decisión (por ejemplo, para una declaración IF, uno para el resultado verdadero y otro para el resultado falso; para una declaración CASE, se necesitarían casos de prueba para todos los resultados posibles, incluido el resultado por defecto).

La cobertura se mide como el número de resultados de decisión ejecutados por las pruebas dividido por el número total de resultados de decisión en el objeto de prueba, normalmente expresado como un porcentaje.

4.3.3 El Valor de la Prueba de Sentencia y Decisión

Cuando se logra una cobertura del 100% de sentencia, se asegura de que todas las sentencias ejecutables del código se han probado al menos una vez, pero no asegura que se haya probado toda la lógica de decisión. De las dos técnicas de caja blanca discutidas en este programa, la prueba de sentencia puede proporcionar menos cobertura que la prueba de decisión.

Cuando se alcanza el 100% de cobertura de decisión, se ejecutan todos los resultados de decisión, lo que incluye probar el resultado verdadero y también el resultado falso, incluso cuando no hay una sentencia falsa explícita (por ejemplo, en el caso de una sentencia IF sin un ELSE en el código). La cobertura de sentencia ayuda a encontrar defectos en el código que no fueron practicados por otras pruebas. La cobertura de decisión ayuda a encontrar defectos en el código donde otras pruebas no han tenido ambos resultados, verdadero y falso.

Lograr una cobertura del 100% de decisión garantiza una cobertura del 100% de sentencia (pero no al revés).

4.4 Técnicas de Prueba Basadas en la Experiencia

Al aplicar técnicas de prueba basadas en la experiencia, los casos de prueba se obtienen a partir de la competencia e intuición del probador y de su experiencia con aplicaciones y tecnologías similares. Estas técnicas pueden ser útiles para identificar pruebas que no fueron fácilmente identificadas por otras técnicas más sistemáticas. Dependiendo del enfoque y la experiencia del probador, estas técnicas pueden lograr grados muy diferentes de cobertura y efectividad. La cobertura puede ser difícil de evaluar y puede no ser medible con estas técnicas.

En las siguientes secciones se abordan las técnicas basadas en la experiencia que se utilizan con frecuencia.

4.4.1 Predicción de Errores

La predicción de errores es una técnica utilizada para anticipar la ocurrencia de equivocaciones, defectos y fallos, basada en el conocimiento del probador, incluido:

- Cómo ha funcionado la aplicación en el pasado.
- Qué tipo de equivocaciones tienden a cometer los desarrolladores.
- Fallos que se han producido en otras aplicaciones.