

La Prueba

La etapa de testing es la adecuada para detectar errores en forma temprana, el éxito de un proyecto de software depende de la misma.

Es catalogado como un elemento secundario en el proceso de software, las organizaciones de desarrollo de software han padecido pérdidas económicas y pérdida de imagen por no realizar un correcto testing de sus aplicaciones.

Durante las últimas tres décadas, el testing ha representado más del 50% del esfuerzo y el costo de los proyectos de desarrollo de además de que organizaciones educativas comenzaron a dictar materias dedicadas exclusivamente a software testing, para que los profesionales de Sistemas tengan conocimiento en la temática.

Las fallas en sistemas cuestan a la economía mundial un estimado de 100 mil millones de dólares anuales, el 35% podría ser eliminado mediante pruebas realizadas durante todas las fases de desarrollo de software.

Es falso decir:

"La prueba es el procedimiento para demostrar que no hay errores en un algoritmo" o "El objetivo de la prueba es demostrar que un programa realiza las funciones previstas de manera correcta" o "La prueba es el proceso para establecer la confianza de que un programa hace lo que se supone tiene que hacer"

Tenemos que pararnos desde la vereda de en frente y plantearlo al revés.

Cuando se prueba un programa se desea agregar valor al mismo, elevar la calidad y confiabilidad del programa utilizando el testing encontrando y eliminando errores.

Entonces no se prueba para demostrar que funciona, sino que hay que empezar con la suposición de que el algoritmo tiene errores y probarlo para encontrar la mayor cantidad posible.

Entonces:

La prueba es el proceso de ejecución de un programa con la intención de encontrar errores.

En resumen, es más apropiado ver al testing como el proceso destructivo de tratar de encontrar errores. Un caso de prueba exitoso es aquel que promueve el progreso en esta dirección, haciendo que el programa falle. Por supuesto, finalmente se desea utilizar el testing para establecer un cierto grado de confianza en que un programa hace lo que se supone que debe hacer y no hace lo que no se supone que debe hacer, pero esto se logra mejor mediante una búsqueda a conciencia de los errores.

Dos de las estrategias más comunes incluyen las pruebas de caja negra (black-box testing) y las pruebas de caja blanca (white-box testing)

Una estrategia de testing importante es la prueba de **caja negra** (black-box, data-driven, ó input/output-driven testing). Para utilizar este método, vea el programa como una caja negra.

No me centro en el comportamiento interno y la estructura del programa, enfocarse en encontrar errores en los que el algoritmo carece de funcionalidad, no se comporta tal lo esperado.

En este enfoque, los datos de prueba se derivan únicamente de las especificaciones (es decir, sin tomar ventaja del conocimiento de la estructura interna del programa).

Si desea utilizar este método para encontrar todos los errores en el programa, el criterio es la prueba exhaustiva de las entradas, generando un caso de prueba para cada de una de las condiciones de entrada posibles.

Esto formará parte de la estrategia de diseño de los casos de prueba abordada más adelante.

Otra estrategia de prueba, prueba de **caja blanca** (white-box ó logic-driven testing), permite examinar la estructura interna del programa. Esta estrategia deriva los datos de prueba de un análisis de la lógica del programa (y, a menudo, por desgracia, ante la omisión en la especificación).

El objetivo es establecer, para esta estrategia, la analogía a la prueba exhaustiva de las entradas del enfoque de caja negra.

Hacer que cada línea en el programa se ejecute al menos una vez puede parecer la respuesta, pero no es difícil demostrar que esto es muy insuficiente.

En éste caso se aplica la prueba de ruta exhaustiva (exhaustive path testing). Es decir, si se ejecutan mediante de casos de prueba todos los posibles caminos (rutas) del flujo de control, entonces, posiblemente, el programa ha sido completamente probado.

Aunque la prueba exhaustiva de las entradas es superior a la prueba de ruta exhaustiva, ninguna resulta ser útil porque ambas son inviables.

Quizás, entonces, haya maneras de combinar elementos de prueba de la caja negra y caja blanca para obtener una estrategia de prueba razonable, pero no infalible.

Hasta aquí se establecieron 3 conceptos introductorios Prueba, Prueba de Caja Negra, Prueba de Caja Blanca.

