

Pruebas de software [resumen]

Las pruebas del software son parte del control de calidad del proyecto, ya que a partir de ellas se pueden verificar la existencia de errores estructurales (codificación) o de diseño en el mismo y se da la oportunidad de hacer las debidas correcciones para entregar como resultado un producto con un alto nivel de confianza y por supuesto, de calidad.

Al probar nuestro software, podemos comprobar que se cumple con todos los requerimientos que se plantearon al inicio del desarrollo de nuestro proyecto.

Las pruebas del software se deben de realizar de acuerdo con un “Plan de pruebas”, en donde especificamos el diseño de las estrategias de caso de prueba y las estrategias de aplicación de las pruebas, así como pasos adicionales y especificaciones de quienes realizaran la pruebas (el desarrollador o un grupo independiente) y la documentación de las mismas para nuestro proyecto.

Para diseñar nuestras estrategias de casos de prueba debemos tener en cuenta lo siguiente:

- Las estrategias de pruebas sirven como guías que definen los pasos a seguir durante el proceso de pruebas de software.
- Debe de existir la garantía de que al realizar las pruebas se detectaran todos los posibles errores que contengan nuestro producto. Además de que la aplicación de estas no deberá de gastar más recurso de los que sean necesarios.
- Las pruebas se deberán de realizar sobre bloques funcionales específicos de nuestro proyecto, los cuales contienen características representativas del mismo, con el objetivo de demostrar que se en tales bloque no hay errores, entonces, en el resto del sistema no habrán.
- Las pruebas son parte del control de calidad (como se señaló anteriormente), por lo que se deben de conocer todos los requerimientos iniciales del sistema al momento de aplicar tales pruebas, ya que a partir de ellos se puede comprobar su correcto funcionamiento.

Dentro del diseño de estrategias de casos de prueba podemos encontrar tres enfoques fundamentales.

- Enfoque estructural: las pruebas son exhaustivas y a nivel de código. Aquí se analizan todas las posibles entradas y salidas proporcionadas por cada instrucción del código fuente.

- Enfoque funcional: aquí las pruebas se realizan para verificar el correcto funcionamiento del sistema, analizando las salidas arrojadas por él, a partir de diversas entradas.
- Prueba aleatoria: se realizan pruebas utilizando entradas basadas en modelos regularmente estadísticos, y se analizan todas las posibles salidas producidas.

La realización de las pruebas se puede llevar el siguiente orden:

- 1) Se realizan pruebas de acuerdo a las estrategias anteriormente planeadas, con la intención de encontrar errores tanto en la estructura interna (código) del sistema o de Diseño, para posteriormente ser depuradas.
- 2) Al finalizar las pruebas se puede decidir entre evaluar los resultados arrojados o llevar a cabo la realización de pruebas adicionales.
- 3) En caso de tener una conclusión exitosa se puede ir directamente a la evaluación de los resultados arrojados
- 4) Si la finalización no es exitosa entonces se recurre nuevamente a la aplicación de estrategias de pruebas, o la realización de pruebas adicionales para localizar los errores y corregirlos.

En las estrategias para la aplicación de las pruebas, podemos encontrar dos enfoques principales:

Pruebas de Unidad: Estas pruebas se realizan en las unidades más pequeñas del sistema, es decir, sus componentes. El objetivo fundamental de tales pruebas es verificar el procesamiento de información dentro de las fronteras del componente, o sea, ¿Cómo opera cada componente? y además ¿Opera correctamente?

Estas pruebas se pueden realizar antes o después de haber generado la codificación correspondiente.

Durante los procedimientos, se requieren del uso de dos programas adicionales más que ayuden a realizar las pruebas correspondientes a cada componente, ya que estos no son programas independientes, y dependen de otros componentes del sistema. Esos dos programas adicionales son el Controlador, quien transfiere la información de entrada que requiere el componente y, un asistente (STUB), quien viene a realizar el trabajo de algún otro componente que este subordinado por el componente en prueba.

Estos programas adicionales (Controlador y asistente), realizan sobrecargas en el proyecto, ya que se codifican de manera independiente a él, y además no se incluyen en la presentación final del mismo.

Pruebas de integración: estas se realizan para enfrentar la problemática en donde, cada componente que se añadirá al sistema opera correctamente *pero solo de manera independiente, porque al ser conjuntado con otras más, comienzan a ocurrir fallos que estropean la integración del sistema.*

Estos tipo de pruebas se realizan en sistemas que se desarrollan de manera incremental, es decir, se crean componentes que conformaran al sistema con el fin de poder aislar con mayor facilidad los errores que se presenten y por ende el tratamiento de tales errores sea más sencillo de lo que sería intentar corregir un sistema no incremental donde el software se conforma como un todo, pero esto hace que al ocurrir algunos errores, la solución de estos sea mucho más complicado y existe el riesgo de que tales errores se propaguen aún más.

Algunas modalidades de realizar este tipo de pruebas son las siguientes:

Descendente: se van probando los componentes desde el más principal hasta el más subordinado. Tras cada iteración de prueba, se deben de sustituir los programas asistentes por componentes reales del sistema. También en este proceso, se realizan pruebas adicionales llamadas de regresión, que se encargan de verificar que no hubo alguna nueva propagación de errores tras la integración de cada componente.

Ascendente: La prueba se realiza desde los componentes de orden más inferior en la jerarquía del sistema hasta llegar al que está en la cima de dicha jerarquía. En este enfoque, ya no son necesarios los programas asistentes, pero si lo programas controladores que brindaran los datos de entrada para cada componente, Al finalizar cada iteración del proceso de prueba, los controladores son sustituidos por componentes reales del sistema.

De regresión: estas pruebas tienen el fin de comprobar que el funcionamiento del sistema no ha sido alterada de manera negativa tras cada modificación que se genera durante las integraciones.

Básicamente La importancia de estas pruebas radica en verificar que no se propaguen nuevos errores en el funcionamiento del sistema cada vez que es agregado un nuevo componente.

De Humo: son utilizadas principalmente en sistemas Críticos. Estas se realizan de manera frecuente y pueden generar las siguientes ventajas:

- Descubren errores de funcionamiento o diseño del sistema, tras cada integración de componentes.
- Facilitan la depuración de dichos errores
- Mejoran la calidad del producto al identificar errores de manera temprana
- Permiten que el progreso del sistema sea favorable para el equipo de desarrollo, ya que las pruebas frecuentes respaldan el buen funcionamiento del sistema tras cada nueva integración de más componentes.

Reportes De pruebas

Se deben de incluir en todo proyecto porque gracias a esta documentación se puede facilitar la detección y corrección de algún error ocurrido durante el proceso de pruebas. Además sirven de evidencia para certificar el funcionamiento correcto de nuestro sistema.

Estos incluyen documentaciones de Entrada, es decir, la documentación de los procesos de prueba que se realizaran; y documentación de salida, o sea, los resultados arrojados durante el proceso de prueba y las correcciones realizadas.

Historial: aquí se registran de manera detallada los eventos más relevantes ocurridos durante el proceso.

Informes de incidente: documenta cada incidente ocurrido en la prueba y que requiera de posterior investigación.

Reporte de Resumen de las pruebas: resume los resultados de las actividades de prueba, y aporta una evaluación del software basada en dichos resultados

Finalmente en el análisis de errores, se hace una compilación de todos los errores registrados las causas que los provocaron.