TAREA N° 10 TESTING

El testing es un proceso “destructivo”, el cual consiste en testear todo el software, una vez realizado a fin de poder encontrar fallas, también conocidos como “bugs”.

Para poder entender que es el testing. Antes debemos entender que es un “Bug”, y para esto debemos saber dónde encontrarlos.   
Un bug, o un fallo, puede ser encontrado en distintos lugares, pero en este caso, que hablamos de Software, nos basamos en Software de aplicación de: Escritorio, Movil y Web.

Una aplicación es un programa que nos permite cumplir con un objetivo. Sin embargo, al momento de realizarlo, se pueden producir errores de distintos tipos. Errores de corrupción de memoria (buffer Overflow), errores que permiten entrar a un disco duro, errores que permiten obtener/modificar cierta información sensible, etc.   
Entonces, al momento de finalizar el desarrollo del Software de Aplicación, hay un sector que se desempeña en probar distintas técnicas de ataques, con el fin de ver dónde se encuentra el fallo, ya que ni un programador debería probar su propio código, ni tampoco la unidad que realizó el software debería hacerlo.   
Cuando se encuentra un fallo, decimos que se encuentra un “bug”.

Hasta hace un tiempo atrás, los fallos, se encontraban a base de “prueba y error”. En la actualidad, se mejoraron las técnicas. El testing se realiza en entornos controlados, con el fin de poder verificar los resultados de salida en base al funcionamiento de la aplicación que realiza con los datos que se ingresan, a fin de verificar tanto los requerimientos funcionales como los no funcionales. Un ejemplo sería probar una dirección física inexistente en una aplicación de taxis, revisar el resultado, para luego compararlo con el resultado de una dirección física existente. Las pruebas se realizan de manera diferente en diferentes contextos. Por ejemplo, la seguridad del software será testeada de forma diferente en un sitio de comercio electrónico, que en uno donde se comparten fotografías. No todos los sistemas de Software llevan el mismo nivel de riesgo y no todos los problemas tienen el mismo impacto cuando se producen. Existen distintos tipos de tests. De operación, de Escala completa, de Performance(o capacidad), de sobrecarga, negativos, basados en requerimientos, ergonómicos, de interfaz de usuario, y de documentación de usuario.   
El testing, o las pruebas, tienen niveles. Unidad; Integración; Sistema; Aceptación.

Primero se testea la unidad(clases, bloques, paquetes de servicios.).

Luego están las pruebas de integración. El propósito de estas pruebas es determinar si las distintas unidades que han sido desarrolladas trabajan apropiadamente juntas. Acá se incluyen las pruebas de paquete de servicio, de subsistemas y el sistema completo. No hay una sola prueba, se realiza en distintos niveles, ya que al combinar las unidades pueden aparecer nuevas falas y la combinación aumenta exponencialmente el número de caminos posibles. De otra forma hay fallas que no podrían detectarse.

Luego vienen las pruebas de aceptación de usuario, donde el usuario determina si se ajusta o no a sus necesidades. La meta es establecer confianza en el sistema.

Las pruebas deben planearse, no hacerse de forma improvisada. El proceso de prueba, que, en gran medida es paralelo con otros procesos, contiene el siguiente conjunto de actividades. Planificación -> Diseño -> Ejecución -> Evaluación y reportes.

La planificación de la prueba, puede comenzar cunado comenzamos el desarrollo de Software. Generalmente, durante los requerimientos, pero no podemos ejecutar nada hasta no comenzar la construcción. Los lineamientos se establecen con anticipación, y, determinando el método y nivel de ambición, se crean las bases de la prueba. También se determinan si las pruebas serán manuales o automáticas a fin de estimar la cant. de recursos que se requerirán.

Se estudian si existen programas de pruebas y datos que puedan usarse, o si deben modificarse o crearse nuevos, para corroborar el grado de cobertura de los test, junto a ello, se mantiene un registro de la prueba.

Diseño. Una vez identificado lo que debe probarse, y estimados los recursos requeridos, se identifican y especifican las pruebas que se harán a nivel funcional, donde también se lleva un registro(o descripción detallada) de la prueba y su propósito de manera general. Cada caso se documentará para facilitar el reuso mencionando: hardware, software, y equipamiento de prueba. Y luego se preparan los reportes para informar los resultados de prueba.

Ejecución de pruebas: Se realizan de forma manual o automatizada y se prueban en paralelo. Se ejecuta una vez realizada la especifiación de pruebas y resportes.

Al finalizar la prueba, se analizan los resultados. Si está aprobado o no. Este análisis resulta en reportes de prueba, los cuales resumen el resultado individual de cada prueba y uno final, los recursos gastados, si hay cuellos de botella o no y si el test está aprobado o no.

Si se detectan fallas, se identifica la razón de la falla. Para encontrarlas hay que resolver si se ha ejecutado correctamente el test, si hay algún falla en los datos o programas de prueba, si la falla es causada por bancos de prueba, o si la falla no es atribuible a la prueba, y en ese caso probar nuevamente. No existe un tiempo estimado para el testing ni para cuando detenerlo. Esto va a depender del nivel de riesgo y costos asociados al proyecto. En la práctica estimar cuando se alcanzará una determinada densidad de defectos, que justifique la finalización de pruebas. Para eso se usan estrategias como “Dejar de probar cuando el producto pasa exitosamente el conjunto de pruebas diseñado (“No Failure”)+ cobertura estructural.

“Good Enough”(Bueno Suficiente): cierta cantidad de fallas no críticas es aceptables).

Defectos detectados es similar a la cantidad de defectos estimados.

Junto al ejecutable, vienen también los IC’s, es decir los Items de Configuración, y una vez lanzado el software al mercado, (cuando ya cumpla con todos los requerimientos y finalmente la aceptación de usuario), con el tiempo, se pueden ir detectando nuevos fallos, y para solucionarlos, los desarrolladores sacan parches que van incluyéndolos en las siguientes variantes. En cada variante, se puede modificar, añadir, o quitar distintos los Ítems de configuración. Estos ítems de configuración, mejoran la usabilidad y la estabilidad del Software requerido.

Echavarría Manuel Tomás