

Ben, entonces, vamos a trabajar sobre algunas definiciones.

¿Qué es el Testing? Según el Instituto de Ingeniería, es una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas. Los resultados de dicha ejecución son observados o registrados y, a partir de los mismos, se realiza una evaluación de algún aspecto del sistema o componente, se entiende, ¿no?

Es lo que veníamos comentando con otra palabra, se hace un análisis, se ven los resultados, se compara resultado obtenido, con resultado esperado.

¿Cuál es el objetivo de las pruebas?, nosotros, ¿para qué probamos?, para encontrar fallos en el software. Nuestro objetivo es diseñar pruebas que muestren diferentes errores con la menor cantidad de tiempo y esfuerzo, cuando menos trabajamos para encontrar más errores, mejor. Y eso tiene mucho que ver también con empezar a probar en etapas más tempranas, no esperar al final y probar todo al final.

Entonces nosotros diseñamos una prueba y eso nos va a dar el resultado esperado. Una vez que nosotros diseñamos la prueba, se ejecuta una entrada válida al sistema y eso nos da un resultado, que es el resultado obtenido. Entonces, se compara el resultado esperado con el resultado obtenido y eso es lo que tenemos "ejecución de la prueba".

Ben, a ver, vamos a hacer una definición de testing más informal, decimos que el testing es probar, vamos a decir, probar es ejecutar un componente software con el objetivo de descubrir fallos y encontrar posibilidades de mejora. El objetivo principal es descubrir fallos, pero el vemos que quizás podemos aportar algo a la mejora del producto, bueno, bienvenido, podemos hacer una sugerencia.

Ben, ¿y cuál es el proceso del testing?, el proceso del testing es el siguiente: nosotros tenemos requerimientos, es decir, hay una necesidad, hay que hacer un desarrollo de software y hay alguien que dice "esto es lo que yo necesito". Entonces, se generan los requerimientos. En algunos momentos, cuando empezamos el trabajo de las áreas del sistema hablamos de los casos de uso, ¿se acuerdan?, bueno esos casos de uso serían una especie de formulación del requerimiento. Tenemos un requerimiento, algo para hacer, esto es lo que tiene que hacer el software, eso es el requerimiento, por un lado. Por otro lado, hay un componente a probar, si hay algo del desarrollo que ya se hizo y ya se puede probar. Entonces, tenemos un componente para probar y lo que el componente tiene que hacer es el requerimiento. Ben, entonces nosotros planteamos una estrategia de pruebas, ¿a?, vamos a definir de qué forma queremos probarlo, ¿está bien?, o sea, por ejemplo, necesitamos un lector de código de barras, necesitamos, estamos trabajando un cajero automático, vamos a necesitar tener un cajero automático para probar. O sea, la estrategia tiene que ver con un montón de cosas, con qué herramientas vamos a utilizar, qué hardware vamos a necesitar, ¿vamos a ejecutar este software?, ¿qué es este software?, ¿ejecuta al mismo, ¿vamos a ejecutarlo o no?, ¿a?, hasta donde lo vamos a llevar. Bueno, todo eso tiene que ver con la estrategia de programas, con temas que ya veremos más adelante.

Entonces, se hace la prueba, ¿a?, la prueba en sí y un análisis, según el análisis de esas pruebas y se hace una comparación del resultado esperado, es decir, de mi análisis surge este componente, si yo le doy está la entrada, tiene que tener tal salida, esto va a ser el resultado que yo esté esperando y de hacer la prueba surge el resultado real, es decir lo que veníamos hablando. Entonces, se hace esa comparación y eso va a dar el resultado de la prueba y esta prueba puede tener dos resultados posibles: o la prueba satisfactoria o sucede un incidente.

Ahora, ¿qué es un incidente?, o sea, ¿es un error o no es un error?.