

INSTITUTO TECNOLÓGICO SUPERIOR ZACATECAS OCCIDENTE



MATERIA: FUNDAMENTOS DE INGENIERÍA DE SOFTWARE

TEMA: MODELOS DE PRUEBAS

DOCENTE:

I.S.C. ERIKA JAZMÍN ROBLES GÓMEZ

ELABORADO POR:

OSIEL CHÁVEZ FLORES

RUBÉN GÓMEZ BARRIENTOS

OSIEL BARRIENTOS RAMÍREZ

FABIÁN ARMANDO HERRERA AVALOS

JESÚS AGUSTÍN JUÁREZ GUERRERO

FECHA: 5 DE DICIEMBRE DEL 2014

INTRODUCCIÓN

Un sistema por muy bien hecho que pudiese llegar a estar siempre estará propenso a fallas, porque las fallas no son solo por parte del equipo donde se encuentra el sistema, estas también son en base a el usuario que maneja el equipo ya que muchas veces por hacer uso inadecuado del mismo este es propenso a fallar.

MODELO DE PRUEBAS DE SOFTWARE.

Error: Es una acción humana que provoca que un software contenga una falta. Un *error* puede significar la existencia de una *falta* en el programa, lo cual hace que el sistema *fallé*.

La prueba es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Una prueba tiene éxito si descubre un error no detectado hasta entonces. Es aplicable a la prueba del Software: el 80% de todos los errores descubiertos durante las pruebas surgen al hacer un seguimiento de sólo el 20% de todos los módulos del programa.

Facilidad de prueba Es simplemente lo fácil que se puede probar un programa de computadora. Como la prueba es tan profundamente difícil, merece la pena saber que puede hacer para hacerlo más sencillo.

Tipos de pruebas

Pruebas de Verificación: Se revisa si el resultado corresponde a la especificación del sistema, es decir, si se está construyendo el sistema de manera correcta.

Pruebas de Validación: Se revisa si el resultado es realmente lo que el cliente quería.

Estas son algunas Técnicas de prueba.

Pruebas de Regresión:

Pruebas de Operación:

Pruebas de Escala Completa

Pruebas de Rendimiento:

En la prueba de flujo de datos Las estrategias de prueba de flujo de datos son útiles para seleccionar caminos de prueba de un programa que contenga sentencias if o bucles anidados.

Prueba de bucles. Técnica de prueba de caja blanca que se centra en la validez de las construcciones de bucles.

Bucles simples Se les debe aplicar el siguiente conjunto de pruebas, donde n es el número máximo de pasos permitidos por el bucle

1. Pasar por alto totalmente el bucle
2. Pasar una sola vez por el bucle
3. Pasar dos veces por el bucle
4. Hacer m pasos por el bucle con $m < n$
5. Hacer $n-1$ y $n+1$ pasos por el bucle

Bucles anidados Si se extendiera el enfoque de los bucles simples a los bucles anidados, el número de posibles pruebas aumentaría geométricamente a medida que aumenta el nivel de anidamiento. Esto llevaría a un número impensable de pruebas.

Bucles concatenados: Se pueden probar mediante el enfoque definido para los bucles simples, mientras cada uno de los bucles sea independiente del resto.

Bucles no estructurados: Esta clase de bucles se debe rediseñar para que se ajusten a las construcciones de programación estructurada.

CONCLUSIÓN

Los modelos de pruebas anteriormente mencionados en este ensayo son ejemplos de a lo que se somete un sistema para verificar su correcto funcionamiento.

Bien sabemos que un sistema es imposible que no tenga fallas, mas sin embargo las pruebas son para ver que tan bien está hecho y poner a trabajar el sistema de una forma un poco más fuerte, para en caso de que llegase a tener un error este pudiera ser corregido antes de poner en uso el sistema, cuando una prueba encuentra un error, significa que prueba dio resultado, cuando no pasa eso el sistema tiene que ser sometido a otras hasta pasar la mayoría de ellas exitosamente.

Entre más pruebas logre validar nuestro sistema esto querrá decir que el mismo está bien hecho y que será difícil que llegue a haber dificultades a plazos cortos, además de que ya hechas gran cantidad de pruebas, el sistema tendrá más actualizaciones de estética que de funcionalidad.