



Ingeniería de Software II (CI-4712)

Fundamentos de Pruebas de Software

Referencias básicas

Ingeniería de software. Un enfoque práctico.
Pressman, R. Quinta edición. Mc. Graw Hill
2002

Ingeniería de software. Sommerville, I.
Séptima edición. Addison Wesley 2005



Prueba

- Proceso de ejecutar software con el propósito de probar su funcionalidad y exactitud
- Se realiza para detectar defectos o estimar la confiabilidad

Niveles

- **Unidad:** Su énfasis está en el funcionamiento apropiado de un programa individual
- **Integración:** Se enfoca en el flujo de datos entre los sistemas.
- **Sistema:** Énfasis en el flujo de datos de un programa a otro dentro del sistema. Los sistemas externos son ignorados.
- **Aceptación:** Aceptación formal de un producto terminado y es realizado por los representantes del negocio.

Pruebas a sistemas

- **Improvisada:** realizar casos de pruebas en el momento, lo más rápido posible.
- **Automatizada:** usa una herramienta de prueba. Ejemplos: SQA, WinRunner, Ms Test y Visual Test

Técnicas de Pruebas

- **Caja Blanca:** Intentan probar tanto del código como sea posible.
- **Caja Negra:** no se preocupan de la estructura del código



Prueba Caja Blanca

- Conocida como Caja de Cristal
- Se basa en el conocimiento de la estructura y de las sentencias del programa y requiere un conocimiento exhaustivo del código del programa.



Prueba Caja Blanca

- Los casos de prueba aseguran que
 - Cada camino independiente en el módulo de software es ejecutado
 - Todos los ciclos iterativos son ejecutados y los límites del cuerpo del ciclo
 - Todas las sentencias de estructuras condicionales son ejecutadas en las condiciones verdaderas y falsas
 - Todas las estructuras de datos usadas en el programa son ejecutadas para chequear correctitud

Prueba Caja Blanca – Prueba de condición

If (num1==num2) ... else if (num1>num2) ...

- Prueba de bifurcación
 - num1=10, num2=10
 - num1=10, num2=15 o num1=15, num2=10
- Prueba de dominio
 - num1=10, num2=10
 - num1=5, num2=15
 - num1=15, num2=5

Prueba Caja Blanca – Prueba de flujo de datos

- Enumerar cada sentencia del programa
- Asegurar que el programa no modifique variables globales
- Asegurar que los subprogramas no modifiquen los parámetros dados por los programas que los llaman
- Considerar una variable “x”
- Localizar el número de sentencia “m” donde se inicializa la variable

Prueba Caja Blanca – Prueba de flujo de datos

- Localizar los números de sentencia “ n_1, n_2, \dots, n_x ” donde se usa la variable “x”
- Definir el conjunto de declaraciones entre los números de declaración “m” y “ n_i ” como cadenas de definición-uso
- El método de prueba de flujo de datos indica que cada cadena de definición-uso debe ser cubierta por cada caso de prueba al menos una vez.

Prueba Caja Blanca – Prueba del bucle

- Seleccionar casos de prueba que **no** ejecuten el bucle
- Seleccionar casos de prueba que ejecuten el bucle **exactamente una vez**
- Seleccionar casos de prueba que ejecuten el bucle **más de una vez** hasta un máximo número de veces
- Seleccionar casos de prueba que verifican si el bucle siempre termina



Prueba Caja Negra

- Conocida como prueba funcional
- Se basa en características funcionales del SW

Prueba Caja Negra – Análisis del valor límite

- Incluya los puntos finales del rango de entrada
- Incluya los valores justo debajo de los puntos finales del rango de entrada
- Incluya los valores apenas sobre los puntos finales del rango de entrada