

Clase 3/10 - Práctico

Testing ---> defectos específicamente de código. DESTRUCTIVO

Proceso destructivo, se trata de encontrar defectos, no errores (un error es algo que está mal hecho, etapa en la que se produce), en cambio el defecto es un error en una etapa posterior a la que se produce, puede ser el mismo error pero encontrado en la etapa de testing es más grave, son más costosos de corregir.

El peor tipo de defecto que podemos tener es **en la etapa de requerimiento**

Requerimientos---análisis---diseño---desarrollo/implementación---prueba---despliegue

En cada etapa se va trabajando con la anterior por lo que los errores se van arrastrando.

Las personas que hacen las pruebas tienen que querer romper el sw.

Las personas que realizan el código no deberían ser las mismas que hacen testing. Hay un conflicto de intereses con lo que plantea SCRAM de que todos tienen que poder hacer todo

El testing NO puede asegurar ni calidad en el software ni software de calidad

CONCEPTOS: DEFECTOS, SEVERIDAD Y PRIORIDAD

Severidad	Prioridad
1 – Bloqueante	1 – Urgencia
2 – Crítico	2 – Alta
3 – Mayor	3 – Media
4 – Menor	4 – Baja
5 - Cosmético	

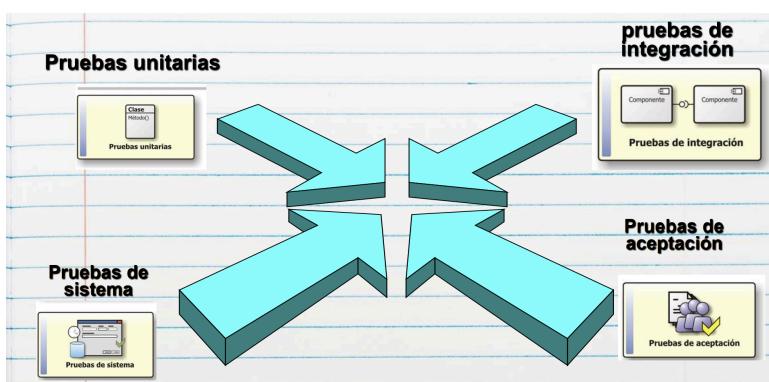
La severidad es que tan grave es el defecto

Lo bloqueante es algo que no me permite utilizar el Sw y lo crítico es algo grave en cuanto a funcionalidades claves pero que sí me permite utilizar el Sw. Los sw tienen un camino que tiene que andar si o si. Una funcionalidad muy importante que no funciona.

La severidad Mayor es algo que no hace a la funcionalidad esencial. Depende de a que esté orientado el SW.

Para la Menor es errores en funcionalidades que se usan poco, menos que las mayor jaja

La prioridad es que tan rápido hay que solucionarlo



Pruebas unitarias funcionan a nivel de código.

NIVELES DE TESTING: TESTING UNITARIO

- Se prueba cada componente tras su realización/construcción.
- Solo se prueban componentes individuales.
- Cada componente es probado de forma independiente
- Se produce con acceso al código bajo pruebas y con el apoyo del entorno de desarrollo, tales como un framework de pruebas unitarias o herramientas de depuración.
- Los errores se suelen reparar tan pronto como se encuentran, sin constancia oficial de los incidentes.

Testing de Integración ---> para identificar cuando las cosas se rompen a la hora de integrarlas

NIVELES DE TESTING: TESTING DE INTEGRACIÓN

- Test orientado a verificar que las partes de un sistema que funcionan bien aisladamente, también lo hacen en conjunto
- Cualquier estrategia de prueba de versión o de integración debe ser incremental, para lo que existen dos esquemas principales:
 - Integración de arriba hacia abajo (top-down)
 - Integración de abajo hacia arriba (bottom-up).
- Lo ideal es una combinación de ambos esquemas.
- Tener en cuenta que los módulos críticos deben ser probados lo más tempranamente posible.
- Los puntos clave del test de integración son simples:
 - Conectar de a poco las partes más complejas
 - Minimizar la necesidad de programas auxiliares

Testing de Sistemas --->

NIVELES DE TESTING: TESTING DE SISTEMA

- Es la prueba realizada cuando una aplicación está funcionando como un todo (Prueba de la construcción Final).
- Trata de determinar si el sistema en su globalidad opera satisfactoriamente (recuperación de fallas, seguridad y protección, stress, performance, etc.)
- El entorno de prueba debe corresponder al entorno de producción tanto como sea posible para reducir al mínimo el riesgo de incidentes debidos al ambiente específicamente y que no se encontraron en las pruebas.
- Deben investigar tanto requerimientos funcionales y no funcionales del sistema.

Testing de Aceptación

TESTING DE ACEPTACION

- Es la prueba realizada por el usuario para determinar si la aplicación se ajusta a sus necesidades.
- La meta en las pruebas de aceptación es el de establecer confianza en el sistema, las partes del sistema o las características específicas y no funcionales del sistema.
- Encontrar defectos no es el foco principal en las pruebas de aceptación.
- Comprende tanto la prueba realizada por el usuario en ambiente de laboratorio (pruebas alfa), como la prueba en ambientes de trabajo reales (pruebas beta).

Clases de Equivalencia

ENTRADAS	ENTRADAS	
Válidas	Inválidas	Condiciones Externas
1.Cadenas de caracteres alfabéticas con 20 caracteres	2. Ingresa numero 3. No ingresa valores 4.Ingresa cadena de más de 20 caracteres	Nombre
5. Ingresa valor numérico de máximo 10 caracteres 6. No ingresa valor	7. Ingresa letras 8. Ingresa más de 10 caracteres	Teléfono

	SALIDA	SALIDA
Condición Externa	Válidas	Inválidas
Mensaje Confirmacion	9. Mensaje logueo exitoso	
Envio de mail	10. Se envia mail informando <>	
Sesión iniciada	11. El usuario se loguea exitosamente	

para el parcial como mínimo 5 clases de equivalencias de error

	Válida	Invalida
Error		12.Error “el nombre solo puede tener letras” 13. Error “nombre obligatorio”