TDA Novena Clase:

Clase de Testing:

Asegurar la calidad no es lo mismo que controlar la calidad, el control de calidad es una de las tareas dentro de lo que respecta al aseguramiento de la calidad. Para ello es necesario tener el claro una serie de consideraciones:

La calidad no puede “inyectarse” al final.

- Lograr un producto de calidad es el resultado de un proceso de calidad, para ello debo controlar este proceso (control de calidad)

- La calidad del producto depende de tareas realizadas durante todo el proceso (inspecciones, auditorias, gestión de configuración, etc.).

- Detectar errores en forma temprana ahorra esfuerzos, tiempo y recursos.

Es el proceso mediante el cual se somete a un software o componente del mismo a condiciones específicas, a fin de determinar y demostrar si es o no válido en función de los requerimientos planteados.

¿Qué principios tiene?

Un programador debería evitar probar su propio código.

- Una unidad de programación no debería probar sus propios desarrollos.

- Examinar el software para probar que no hace lo que se supone que debería hacer es la mitad de la batalla, la otra mitad es ver que hace lo que no se supone que debería hacer.

- No planificar el esfuerzo de testing sobre la suposición de que no se van a encontrar defectos.

- El testing es una tarea extremadamente creativa e intelectualmente desafiante.

Entre otros mas…., *mirar filmina de la profe de clase.

¿Cuánto testing es necesario?

Decidir cuánto testing es suficiente depende de:

o Evaluación del nivel de riesgo

o Costos asociados al proyecto

Usamos los riesgos para determinar:

o Que testear primero

o A qué dedicarle más esfuerzo de testing

o Que no testear (por ahora)

Criterios de Aceptación: El criterio de aceptación es lo que comúnmente se usa para resolver el problema de determinar cuándo una determinada fase de testing ha sido completada. Estos pueden estar definidos en término de:

- Costos
- % de tests corridos sin fallas
- Fallas predichas que aún permanecen en el software
- No hay defectos de una determinada severidad en el software.
- Pasa exitosamente el conjunto de pruebas diseñado y la cobertura estructural.
- Good Enough: Cierta cantidad de fallas no críticas es aceptable.
- Defectos detectados es similar a la cantidad de defectos estimados.

Diferencia entre Error y Defecto

Error es una falla que se identifica en la misma etapa que donde se originó, es decir originada en el producto de trabajo que se está inspeccionando.

Defecto es aquella falla que se identifica en una etapa posterior a aquella donde se originó, es decir es la falla que se trasladó a una etapa posterior, y por lo tanto es un problema mayor.

Proceso de Testing:

Planificación y control de resultados:

La Planificación de las pruebas es la actividad de verificar que se entienden las metas y los objetivos del cliente, las partes interesadas (stakeholders), el proyecto, y los riesgos de las pruebas que se pretende abordar.

Análisis y diseño de las pruebas:

Aquí se toman las entradas, se hace el análisis y se escribe el caso de prueba. Esta etapa incluye las siguientes actividades:

Revisión de la base de pruebas

- Verificación de las especificaciones para el software bajo pruebas
- Evaluar la testeabilidad de los requerimientos y el sistema
- Identificar los datos necesarios.

Ejecución

Esta es la etapa en donde se lleva adelante la ejecución de los casos de prueba en sí mismo, esta incluye las siguientes actividades:

- Desarrollar y dar prioridad a nuestros casos de prueba

- Crear los datos de prueba
- Automatizar lo que sea necesario; para ello hay que analizar el tipo de prueba y los costos (Ej. Las pruebas de interfaz, las de performance, y de stress no se automatizan).

Evaluación y Reporte

En función de los resultados obtenidos luego de la ejecución de los casos de prueba, en esta etapa se debe:

- Evaluar los criterios de Aceptación
- Reporte de los resultados de las pruebas para los stakeholders.
- Recolección de la información de las actividades de prueba completadas para consolidar

Niveles de Testing:

Testing Unitario

- Se prueba cada componente en forma independiente luego de su construcción.
- Generalmente la realizan los desarrolladores.
- 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

Test orientado a verificar que las partes de un sistema que funcionan bien aisladamente, también lo hacen en conjunto. Toda 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)