

## **Pruebas de Software (Software Testing) y Aseguramiento de la calidad QA (siglas de las palabras en inglés Quality Assurance).**

En el modelo Cascada, se realizan las pruebas una vez terminada la construcción del sistema, en el Incremental se realizan las pruebas en cada incremento del sistema, o por ejemplo, en el Evolutivo mediante la retroalimentación de los usuarios, en el espiral durante su verificación y validación del desarrollo, o en los enfoques XP (eXtreme Programming o Programación Extrema) con repetidas pruebas de cada una de las mejoras debido a su desarrollo iterativo e incremental.

En el caso de QA de software, se referirá entonces, a asegurar la calidad de los resultados de cada una de las fases del ciclo de vida del software.

En la primera etapa, Análisis de Requisitos, dentro de un modelo cascada, se ha definido un tipo determinado de documento a presentar, entonces para pasar a la fase de Diseño, el documento de Análisis deberá estar conforme al documento estándar definido por la organización.

La idea es que mientras más temprano se detecten las fallas, menor será el costo (monetario, de tiempo, recursos, calidad, etc.) de repararlas y mayor la calidad del producto final.

### **Software Testing**

#### **Prueba Unitaria**

Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido. Esta prueba se enfoca en comparar el resultado esperado con el resultado obtenido.

#### **Prueba de Integración**

Su objetivo es identificar errores introducidos por la combinación de programas probados unitariamente.

#### **Prueba de Regresión**

Determinar si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes. La prueba de regresión es una nueva corrida de casos de prueba previos.

#### **Pruebas de Humo (Smoke Testing o Ad Hoc)**

Detectar los errores en releases tempranos y de manera fácil, además de probar el sistema constantemente. Este tipo de prueba es útil para realizar una integración de todo el sistema cada cierto periodo (se recomienda un día, máximo una semana).

#### **Pruebas del Sistema**

Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. Se recomienda que en este tipo de prueba se ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que los resultados esperados ocurren cuando se utiliza un dato válido, los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido y por último, cada regla de negocios es aplicada adecuadamente.

#### **Pruebas de Desempeño**

El objetivo de esta prueba es el de validar el tiempo de respuesta para las transacciones o funciones de negocios.

### **Pruebas de Carga**

Este tipo de pruebas se encarga de verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga. Se recomienda, para llevar a cabo este tipo de pruebas, que se modifiquen archivos de datos (para incrementar el número de transacciones o veces que cada transacción ocurre).

### **Pruebas de Stress**

El objetivo de la prueba es verificar que el sistema funciona apropiadamente y sin errores, bajo estas condiciones de stress, es decir, Memoria baja o no disponible en el servidor. Máximo número de clientes conectados o simulados (actuales o físicamente posibles). Múltiples usuarios desempeñando la misma transacción con los mismos datos. Y por último, el peor caso de volumen de transacciones.

### **Pruebas de Volumen**

Verificar que la aplicación funciona adecuadamente bajo escenarios de volumen.

En este tipo de pruebas se deben usar múltiples clientes, ya sea corriendo las mismas pruebas o pruebas complementarias para producir el peor caso de volumen por un período extendido.

### **Pruebas de Recuperación y Tolerancia a fallas**

El objetivo de esta prueba es verificar que los procesos de recuperación (manual o automática) restauran apropiadamente la Base de datos, aplicaciones y sistemas, y los llevan a un estado conocido o deseado. Considere,

- Interrupción de electricidad en el cliente.
- Interrupción de electricidad en el servidor
- Interrupción en la comunicación hacia el servidor (caídas de red)
- Interrupción en la comunicación con los controladores de disco.
- Ciclos incompletos (procesos de consultas interrumpidos, procesos de sincronización de datos interrumpidos)
- Llaves o apuntadores de base de datos inválidos
- Elementos corruptos o inválidos en la base de datos.

### **Prueba de Múltiples Sitios**

El principal objetivo de esta prueba es detectar fallas en configuraciones y comunicaciones de datos entre múltiples sitios.

### **Prueba de Compatibilidad y Conversión**

Su objetivo es buscar problemas de compatibilidad y conversión en los sistemas.

### **Pruebas de Integridad de Datos y Base de Datos**

Esta prueba busca asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos.

Se recomienda que usted invoque cada método de acceso y proceso de la Base de datos, utilizando en cada uno datos válidos e inválidos.

### **Pruebas de Seguridad y Control de Acceso**

Se debe medir el nivel de seguridad de la aplicación, es decir, Verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.

También se debe probar el nivel de Seguridad del Sistema, esto consiste en verificar que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

### **Pruebas del Ciclo del Negocio**

Prueba que se encarga de asegurar que el sistema funciona de acuerdo con el modelo de negocios, emulando todos los eventos en el tiempo y en función del tiempo.

### **Pruebas de GUI**

Las navegaciones a través de los objetos de la prueba reflejan las funcionalidades del negocio y requisitos, para esta prueba se realiza una navegación ventana por ventana, usando los modos de acceso (tabuladores, movimientos del mouse, teclas rápidas, etc)

Los objetos de la ventana y características, tales como menús, medidas, posiciones, estados y focos se verifican conforme a los estándares.

### **Pruebas de Configuración**

Validar y verificar que el cliente del sistema funciona apropiadamente en las estaciones de trabajo recomendadas.

### **Prueba de Estilo**

Comprobar que la aplicación sigue los estándares de estilo propios del cliente.

Se realiza una navegación por la aplicación verificando si se cumplen con los estándares de GUI del cliente.

### **Prueba de Aceptación**

Determinación por parte del cliente de la aceptación o rechazo del sistema desarrollado.

### **Prueba de Instalación**

Verificar y validar que el sistema se instala apropiadamente en cada cliente.

### **Pruebas Funcionales**

En esta prueba se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. En este proceso se ejecuta cada caso de uso, flujo de caso de uso, o función, usando datos válidos e inválidos.

### **Prueba de Documentación Y Procedimiento**

Evaluar la exactitud y claridad de la documentación del usuario y para determinar si el manual de procedimientos trabajará correctamente como una parte integral del sistema.

### **Prueba de Usabilidad**

Determina cuán bien el usuario podrá usar y entender la aplicación. Identifica las áreas de diseño que hacen al sistema de difícil uso para el usuario.

Poner atención en:

- El sistema es demasiado complejo y difícil de usar.
- Es difícil instalar y entender el sistema
- La recuperación de errores es pobre y los mensajes de error no tienen significado
- La sintaxis de los comandos es difícil de aprender y recordar
- El sistema obliga al usuario a recordar formatos y secuencias fijas
- Los procedimientos no son simples ni obvios
- El sistema no tiene instrucciones de ayuda por computadora y tiene manuales pobres.
- Los diagramas, pantallas, reportes y gráficos son de calidad y apariencia pobre
- El sistema carece de herramientas de construcción adecuadas y requiere múltiples comandos
- La lógica y conveniencia de los botones, switches, displays y mensajes de ayuda deben ser testeados. (La prueba de usabilidad puede ser conducida por un grupo separado si es posible.

### **Prueba de Campo**

Correr el sistema en el ambiente real para encontrar errores y validar el producto contra sus especificaciones originales.

### **Pruebas Alfa**

Prueba de aceptación para detectar errores en el sistema bajo un ambiente controlado.

Se llevan a cabo en el lugar en donde fue desarrollado el SoftWare, en un ambiente controlado, en el cual el desarrollador está presente.

### **Pruebas Beta**

Prueba de aceptación donde La validación (o pruebas beta) involucra el uso del software en un ambiente real.

Se selecciona un grupo de usuarios que ponen a trabajar el sistema en un ambiente real. Usan el sistema en sus actividades cotidianas, procesan transacciones y producen salidas normales del sistema.