



El test estático es una técnica de test de software que se utiliza para verificar defectos en la aplicación del software sin ejecutar el código. Las pruebas estáticas se realizan para evitar errores en una etapa temprana de desarrollo, ya que de este modo, es más fácil identificar los errores y resolverlos.

También ayuda a encontrar errores que Dynamic Testing no puede encontrar. Siendo el test dinámico la contraparte, verifica una aplicación cuando se ejecuta el código.

Tipos de técnicas principales de pruebas estáticas

Exámenes manuales: Los exámenes manuales incluyen el análisis del código realizado de forma manual, también conocido como Revisiones.

Análisis automatizado con herramientas: El análisis automatizado es básicamente un análisis estático que se realiza con el uso de herramientas.

¿Qué es la revisión de pruebas?

Una revisión en un test estático es un proceso o reunión realizada para encontrar los defectos potenciales en el diseño de cualquier programa. Otro significado de la revisión es que todos los miembros del equipo conocen el progreso del proyecto y a veces, la diversidad de pensamientos puede resultar en excelentes sugerencias. Los documentos son examinados directamente por personas y las discrepancias se pueden resolver.

Las revisiones se pueden clasificar en cuatro partes:

- Revisiones informales
- Tutoriales
- Revisión técnica
- Inspecciones

Durante el proceso de revisión, los cuatro tipos de participantes que toman parte en las



pruebas son:

- Moderador: Realiza la verificación de entrada, da seguimiento al retrabajo, miembro del equipo de entrenamiento, programa la reunión.
- Autor: Se responsabiliza de corregir el defecto encontrado y mejora la calidad del documento.
- Scribe: Realiza el registro del defecto durante una revisión y asiste a la reunión de revisión.
- Revisor: Verifica el material en busca de defectos mediante inspecciones.
- Gerente: Decide sobre la ejecución de revisiones y asegura que se cumplan los objetivos del proceso de revisión.

Algunos de los defectos que son más fáciles de encontrar durante las pruebas estáticas son:

- Desviaciones de los estándares
- Código no mantenible
- Defectos de diseño
- Requisitos faltantes
- Especificaciones de la interfaz inconsistentes

Por lo general, el defecto descubierto durante las pruebas estáticas se debe a vulnerabilidades de seguridad, variables no declaradas, violaciones de límites, violaciones de sintaxis, interfaz inconsistente, entre otros.

Por qué hacer pruebas estáticas?

Algunas de las principales razones para realizar pruebas estáticas son:

- Detección y corrección temprana de errores
- Plazos de desarrollo reducidos
- Costo y tiempo de prueba reducidos
- Para mejorar la productividad del desarrollo



- Para obtener menos defectos en una etapa posterior de prueba

Qué tests se realizan en las pruebas estáticas?

Al hacer pruebas estáticas, se prueba lo siguiente:

- Casos de prueba unitaria
- Documento de requisitos comerciales (BRD)
- Casos de uso
- Requisitos del sistema/funcionales
- Prototipo
- Documento de especificación del prototipo
- Hoja de cálculo del diccionario de campos DB
- Datos de prueba
- Documento Matriz de Trazabilidad
- Manual de usuario, guías de formación, documentación
- Documento de estrategia del plan de prueba
- Scripts de prueba de automatización/rendimiento

Cómo se realizan las pruebas estáticas

Para hacer pruebas estáticas, se puede seguir lo siguiente:

- Llevar a cabo el proceso de inspección para conocer a fondo el diseño de la aplicación.
- Usar una lista de verificación para cada documento bajo revisión para asegurarse de que todas las revisiones estén cubiertas completamente.

Las actividades para realizar tests estáticos son:

1.- Validación de requisitos de casos de uso: Valida que se identifiquen todas las acciones del usuario final, así como cualquier entrada y salida asociada con las mismas. Cuanto más



detallados y completos sean los casos de uso, más precisos y completos pueden ser los casos de prueba.

2.- Validación de requerimientos funcionales: Asegura que los requerimientos funcionales identifiquen todos los elementos necesarios. También analiza la funcionalidad de la base de datos, las listas de interfaces y los requisitos de hardware, software y red.

3.- Revisión de arquitectura: Todos los procesos de nivel comercial, como ubicaciones de servidores, diagramas de red, definiciones de protocolos, balanceo de carga, accesibilidad a bases de datos, equipos de prueba, etcétera.

4.- Validación de prototipo/maqueta de pantalla: Esta etapa incluye la validación de requisitos y casos de uso.

5.- Validación del diccionario de campos: Cada campo en la interfaz de usuario se define lo suficientemente bien como para crear casos de prueba de validación a nivel de campo. Los campos comprueban la longitud mínima/máxima, los valores de lista, los mensajes de error, etcétera.