

Estrategia de Pruebas para la Aplicación Swag Labs

1. Estrategias

1.1. Entendimiento:

Se debe realizar procesos de contextualización del proyecto o partes de funcionales identificando los usuarios y sus necesidades que se desean para los procesos de pruebas, con un tiempo estimado y alcances definidos para poder validar estimaciones de casos pruebas y esfuerzos.

1.2. Definición de objetivos de las pruebas:

Establecer claramente los objetivos que deseas lograr para realizar las pruebas, definiendo criterios de éxito para evaluar el rendimiento del software.

1.3. Selección de Tipos de pruebas:

Pruebas Unitarias:

Verificar que cada componente individual de la aplicación funcione correctamente.

Pruebas de Integración:

Asegurar que los módulos de la aplicación trabajen de manera conjunta sin problemas.

Pruebas de Sistema:

Evaluar el rendimiento y la funcionalidad de la aplicación en su conjunto.

Pruebas de Aceptación del Usuario (UAT):

Validar que la aplicación cumple con las expectativas de los usuarios finales.

Pruebas Funcionales:

Confirmar que cada función de la aplicación realiza la tarea prevista.

Pruebas de Regresión:

Asegurar que nuevas actualizaciones no afecten a las funciones existentes.

Pruebas de Rendimiento:

Evaluar la capacidad de la aplicación bajo diferentes condiciones de carga.

Pruebas de Seguridad:

Identificar y corregir posibles vulnerabilidades de seguridad.

1.4. Herramientas de pruebas

Selecciona las herramientas de pruebas adecuadas para el proyecto. Automatiza pruebas cuando sea posible para aumentar la eficiencia.

1.5. Ambiente de pruebas

Configurar un entorno de pruebas que sea similar al entorno de producción. Asegurándose de que el entorno de pruebas sea consistente y replicable

1.6. Casos de pruebas

Crea casos de prueba detallados para cada funcionalidad. Incluyendo casos para condiciones límite, escenarios de error y casos positivos/negativos

1.7. Pruebas de rendimiento

Las pruebas de rendimiento para evaluar la capacidad del sistema bajo carga. Ejecutar pruebas de seguridad para identificar y corregir posibles vulnerabilidades.

1.8. Gestión de defectos.

Sistema que permite realizar seguimientos, gestionar defectos, priorizar y aborda los defectos de manera oportuna.

2. Procesos de seguimiento de bug's

Los procesos de seguimiento de bug son fundamentales para garantizar la calidad, la eficiencia y la satisfacción del usuario en el desarrollo de software. A continuación, se detalla el paso a paso del proceso de seguimiento.

2.1. Identificación del Bug:

Proporcionar información detallada sobre el error, incluyendo cómo reproducirlo, mensajes de error, y cualquier otra información relevante.

2.2. Registro del Bug:

Crea un registro de bug en un sistema de seguimiento de problemas (como Jira, Redmine, Trello, YODIZ, etc.).

Incluye detalles como la descripción del bug, pasos para reproducir, entorno en el que se encontró, y su gravedad/prioridad.

2.3. Asignación del Bug:

Asigna el bug al miembro del equipo responsable de abordar el problema. Esto podría ser un desarrollador específico o un miembro del equipo.

2.4. Priorización del Bug:

La asignación de prioridad de los bugs es basada en su gravedad y en su impacto en el software. Esto ayuda a determinar qué bugs deben ser abordados primero.

2.5. Reproducción y Confirmación del Bug:

El desarrollador o el tester debe intentar reproducir el bug en su propio entorno para comprender mejor la naturaleza del problema también validando que la información proporcionada es suficiente para entender y solucionar el problema.

2.6. Corrección del Bug:

La persona asignada trabaja en corregir el código para abordar el bug. Este proceso puede implicar pruebas unitarias para garantizar que la corrección no introduzca nuevos problemas.

2.7. Pruebas de Validación:

Después de corregir el bug, el equipo de calidad realiza pruebas de validación o regresión para confirmar que el problema se ha solucionado y que la corrección no ha introducido nuevos bugs.

2.8. Actualización del Estado del Bug:

Actualiza el estado del bug en el sistema de seguimiento para reflejar su progreso (por ejemplo, de "abierto" a "corregido" o "cerrado").