

SOPORTE DE HERRAMIENTAS PARA EL PROCESO DE PRUEBAS

- Consideraciones para el proceso de prueba
 - Herramientas de ejecución: SELENIUM
 - Herramientas de soporte: GenerateData

Las herramientas de prueba se utilizan para evitar la monotonía al realizar una y otra vez las mismas pruebas y ahorra tiempo de trabajo.

- Clasificación
 - Objetivo
 - Precio
 - Tipo de licencia
 - Tecnología
 - Soporte que brinda

Existen diferentes tipos de herramientas que pueden incluir varios tipos de programas que facilite el trabajo para el desarrollo del software.

- Efecto sonda (probe effect) El efecto sonda es el que produce el instrumento que mide, sobre el objeto que se está midiendo. Ejemplo Un aparato que mide la velocidad del corredor, pero el aparato tiene que ser cargado por el corredor y el peso extra del aparato afecta el desempeño del mismo.

Si se quiere medir la cobertura de decisión sobre una herramienta, pero la herramienta debe de interactuar sobre lo que se está probando el intercambio de información entre ambos puede afectar el tiempo de respuesta. También puede aparecer un bug cuando se está revisando pero al utilizar la herramienta el bug ya no aparece, esto por la manera en cómo afecta la memoria.

- Tesware. Software diseñado para ejecutar pruebas automáticamente.

Tipos de herramientas:

- 1- Gestión de pruebas
- 2- Pruebas estáticas
- 3- Diseño e implementación
- 4- Ejecución y registro
- 5- Medición de rendimiento
- 6- Análisis dinámico
- 7- Pruebas especializadas

GESTION DE PRUEBAS

- 1- Soporte de herramientas para la gestión de prueba y productos.
 - a. Herramientas para la gestión de la prueba y ciclo de vida. Apoyan la gestión de reportes de incidentes y estados del proceso de pruebas. (capturas de pantallas, reportes de bloqueo, de seguimiento, de impacto).
 - b. Herramienta de gestión de defectos también conocido como gestión de incidentes. Datos corruptos, errores del probador, solicitud de mejora

- c. Herramienta para gestión de requisitos. Garantiza trazabilidad de los requisitos así como la cobertura de estos en las pruebas
 - d. Herramientas de gestión de la configuración. Tener el control sobre lo que estamos probando.
 - e. Herramientas de integración continua. Debox (personal de desarrollo).
- 2- Soporte de Herramientas para pruebas estáticas.
 - a. Herramientas para revisión. Informales es compartirlo con compañeros para que revisen la documentación. Generalmente este tipo de herramientas contiene un ID, comentarios, alertas, agenda y coordinación, defectos y estadísticas, trazabilidad, bases de datos y monitoreo de revisión.
 - b. Herramientas de análisis estático (Desarrolladores) Buscador con palabras claves, comprensión del código, complejidad ciclomatica, validación de estándares, análisis de estructura.
- 3- Soporte de herramientas para Diseño e implementación de pruebas. Herramientas de soporte como modelado.
 - a. Creación de casos y procedimientos y datos de prueba
 - b. Resultados esperados
 - c. Datos de entrada
 - d. Interacción con UI
 - i. Herramientas de medición de cobertura de la prueba. Cuantas coberturas se deben realizar.
 - e. Herramientas basadas en modelos. Generar datos o casos a partir de modelos
- 4- Herramientas de Preparación de datos de prueba. Extraer datos de bases de datos, los configura u ordena,
- 5- Herramientas de Aceptación. Se dividen en 3:
 - a. Test Driven Development TDD. Se crean casos de prueba desde la perspectiva del desarrollador **es decir de adentro hacia afuera**. El probador crea casos de pruebas para contestar la pregunta de si el código es correcto.
 - b. Behavior Driven Development BDD. Desarrollo guiado por el comportamiento. **Se analiza desde afuera hacia adentro**. Se crean los datos de prueba con la participación de los desarrolladores, probadores y usuarios. La pregunta que debe hacerse **¿es esto lo que deberíamos probar?**
 - c. Acceptance Testing Driven Development. ATDD. Se crean los casos de prueba desde la perspectiva del negocio. La pregunta es **¿El código hace lo que se espera?**
- 6- Soporte de Herramientas para la ejecución y registro de Prueba
 - a. Herramientas para la ejecución de pruebas nos sirve para ejecutar y guardar datos de prueba mientras se ejecutan manualmente (grabar mientras se ejecuta la prueba manualmente). Esto para generar estadísticas que se puedan realizar más adelante. Otro ejemplo son los arneses de prueba o Testing Frameworks que son aquellos conjuntos de elementos que permiten ejecutar una prueba automáticamente.

- i. Herramientas de cobertura. Permiten instrumentar el código y calcular el porcentaje de cobertura.
- 7- Soporte de Herramientas para medir rendimiento y análisis dinámico. Simular situaciones del entorno real donde funcionara el sistema.
 - a. Herramientas para pruebas de rendimiento. No funcionales y miden el tiempo de respuesta en el mundo real para conocer la capacidad del sistema y sus límites. Se dividen en tres
 - i. Carga. Se ejecutan para monitorear el comportamiento de un sistema con un flujo de usuarios simultáneos.
 - ii. Estrés. Forzar al sistema hasta donde los datos de carga y tiempo de respuestas no satisfacen las expectativas.
 - iii. Estabilidad. Miden la estabilidad del sistema bajo prueba. Por ejemplo que pasa si se corta la conexión con internet.
 - b. Herramientas de monitorización. Cantidad de usuarios en el sistema y registrar toda la data. Se usan principalmente para monitorear servidores o bases de datos.
 - c. Herramientas de análisis Dinámico. Analizan comportamiento del software mientras se está ejecutando como tiempo de respuesta, fallas aritméticas etc. Es decir, sirven para evaluar que tan bien ejecuta el sistema bajo prueba lo que se supone **que debe hacer y el ¿cómo lo hace? mientras el ¿qué hace? queda fuera.**
- 8- Soporte de Herramientas para necesidades de Pruebas especializadas. Para cierto tipo de software ya que no es lo mismo el sistema de robustez de IOS que de ANDROID o un sistema de rendimiento aeronáutico.

Beneficios y riesgos de la automatización de la prueba.

- Beneficios
 - Menos trabajo repetitivo
 - Mayor consistencia
 - Evaluaciones de pruebas más objetivas
 - Información centralizada
 - Ejemplo: Un beneficio es ya que cada que se hace una modificación del software se tiene que volver a realizar todas las pruebas, es por ello que se puede automatizar ya que la monotonía comienza a ocasionar errores humanos en las pruebas, pero esto no es siempre una buena opción ya que existe la resistencia al cambio.
- Riesgos
 - Expectativas no satisfechas (tiempo, costo y capacitación del personal antes de tener resultados rentables).
 - Subestimación del esfuerzo
 - Responsabilidades no claras (Proveedor)
 - Fallas en la Interoperabilidad (que las herramientas no se llevan entre si o se ve afectada al utilizarlas en conjunto)

Consideraciones Especiales con respecto a las herramientas de Ejecución y Gestión de Prueba.

- Funcionan ejecutando el software usando un script
- Tipos de SCRIPTS
 - Lineales. Manuales donde se escribe el código
 - Estructurados. If, switch, While (condicionales)
 - Compartidos. Donde varias secuencias de comandos pueden llamarse entre ellas para utilizarse
 - Data-driven Scripts. Datos y resultados esperados se almacenan por separado el script es más genérico.
 - Keyword-Driven Script. Archivo donde con palabras clave entra una acción que llama a los scripts respectivos de acuerdo a las palabras claves.

Elección de una Herramienta

- Generalmente se utiliza un proyecto piloto o un proyecto pequeño para utilizar la herramienta para ver cómo se va a desempeñar y conocer como actuara en futuros proyectos.
- Identifica la necesidad y se busca una herramienta que satisfaga dicha necesidad.
- Nunca se van a cambiar mis procesos solo para utilizar un tipo de herramienta.
- Validar el tipo de licencia que tiene la Herramienta
- Tiempo de capacitación del personal
- Condiciones del contrato de la herramienta
- Evaluar al proveedor de la herramienta
- Comparar la mejor opción