



Tecnológico de Monterrey

Integración de seguridad informática en redes
y sistemas de software
(Gpo 401)

Pruebas de software

Integrantes:

Joshua R. Amaya Camilo		A01025258
Octavio Fenollosa Sosa		A01781042
Ariadne Álvarez Reyes		A01652080
Andrea Serrano Diego		A01028728

Profesores:

Gilberto Echeverría Furió
Federico Navarro Bernáldez
Jorge Rodríguez Ruíz

Fecha de entrega: Jueves 25 de Agosto de 2022

Pruebas de software

Durante el proceso de desarrollo de software, es común encontrar errores. Cuando esto sucede, se debe hacer una serie de pruebas de software para su verificación y solución de la funcionalidad. Esto ayuda a dar una garantía de calidad sobre el sistema y su correcto funcionamiento. La intención final es que coincida con los requisitos esperados para entregar un producto de calidad. Implica la ejecución de componentes de software o sistema utilizando herramientas manuales o automatizadas para evaluar una o más propiedades de interés. El testing de software es un proceso paralelo al desarrollo de software cuyas tareas deben ir realizándose a medida que se construye el producto para evitar problemas en la funcionalidad de manera previa a su lanzamiento.

Las pruebas de software son importantes porque permiten identificar de manera temprana si hay algún problema en el mismo, facilitando su resolución antes de la entrega del producto. Un producto de software debidamente probado garantiza calidad, seguridad, confiabilidad y alto rendimiento, además de otros beneficios como ahorro de tiempo, seguridad y satisfacción del cliente.

Los tipos de pruebas se dividen en:

Pruebas funcionales

Verifican cada función de una aplicación o software, su funcionalidad con un conjunto específico de requisitos.

Pruebas no funcionales o pruebas de rendimiento

Consideran parámetros como la confiabilidad, la usabilidad y el rendimiento.

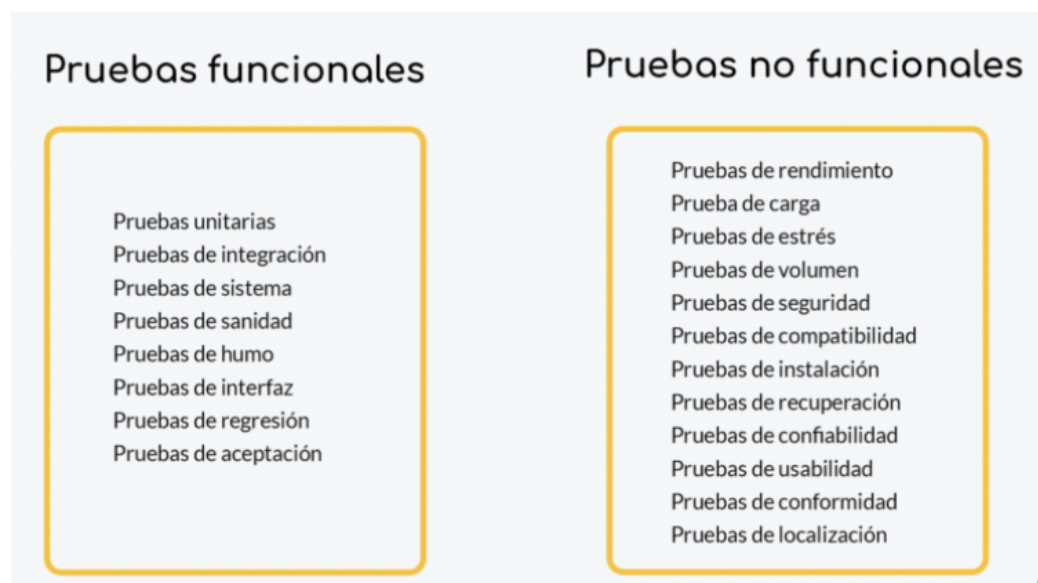


Imagen 1: Tipos de pruebas de software

Pruebas funcionales

Estas pruebas se aseguran de que el sitio web / aplicación está libre de defectos. Garantizan el comportamiento esperado de todas las funcionalidades y que la arquitectura sea correcta con la seguridad necesaria. Por otro lado, mejora la calidad y las funcionalidades generales y minimiza los riesgos empresariales asociados con el sistema.

Pruebas de integración.

- Se asegura de que todos los módulos de aplicación estén bien integrados y funcionen juntos según lo esperado.
- Detecta problemas y conflictos interconectados para resolverlos antes de crear un gran problema.
- Válida la funcionalidad, fiabilidad y estabilidad entre diferentes módulos.
- Detecta excepciones ignoradas para mejorar la calidad del código.
- Admite la canalización de CI/CD.

Pruebas de regresión.

Las pruebas de regresión verifican un conjunto de escenarios que funcionaron correctamente en el pasado, para asegurar que continúen así.

No se deben agregar nuevas características al regression test suite hasta que las pruebas de regresión actuales pasen.

Una falla en una prueba de regresión significa que una nueva funcionalidad ha afectado otra funcionalidad que era correcta en el pasado, causando una "regresión".

Una falla en un test de regresión podría indicar también se ha vuelto a producir un bug que ya había sido resuelto en el pasado.

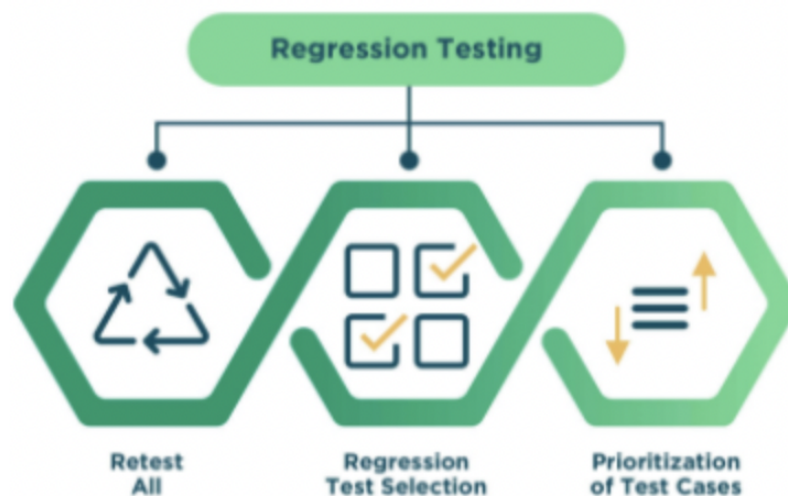


Imagen 2: diagrama de pruebas de regresión

Pruebas locales

Las pruebas locales dan flexibilidad de poder ejecutar las aplicaciones que se están desarrollando en entornos locales, como un ordenador, servidores o sistemas de integración continua, en un entorno parecido al de producción, pero sin el tiempo y el proceso de despliegue

Pruebas alfa

Las pruebas alfa son las que se ejecutan en las oficinas del desarrollador del producto por un grupo de personas que representan al cliente final. En estas pruebas, el desarrollador estará junto a estos usuarios registrando errores y problemas de uso.

Por otro lado, también tenemos las pruebas beta, que se realizan en las oficinas de un cliente o en varias situaciones específicas, ya que el usuario final puede ser varios clientes a los que se entrega el producto. Durante estas pruebas el desarrollador no está presente y los clientes registran todos los problemas derivados del uso de este producto, lo que después se hacen llegar al desarrollador.

Pruebas dinámicas de validación

Las pruebas dinámicas de validación se describen como una especie de software de prueba que analiza el comportamiento dinámico del código. Este tipo de prueba se utiliza para monitorear el comportamiento funcional del software y verificar que cumpla con los requisitos y expectativas del cliente. De igual manera, le permite al equipo responder a problemas respecto a cómo se cumple el producto de software y el cómo se desempeña en un ambiente externo.

Los aspectos principales que se evalúan en las pruebas dinámicas es garantizar el correcto funcionamiento de la aplicación desarrollada, ya sea durante o después de la instalación de la misma; asegurar la estabilidad y eficiencia en la ejecución del software, ya sea como resultado o como proceso.

Pruebas bajo condiciones frontera.

Estas pruebas son el análisis de valores límite que se basa en la comprobación de los valores límite de las particiones válidas e inválidas. Es más probable que el comportamiento en el borde de la partición de equivalencia sea incorrecto que el comportamiento dentro de la partición, por lo que los límites son un área en la que las pruebas pueden producir defectos.

Se comprueban los valores de entrada cerca del límite que tienen una mayor probabilidad de error. Cada partición tiene sus valores máximos y mínimos.

Ejemplos de herramientas para pruebas de automatización

El objetivo de estas herramientas es la creación de scripts en diferentes lenguajes de programación, dependiendo de la herramienta que se vayan a utilizar, que permitan ejecutar las pruebas funcionales automáticamente.

Selenium (libre): Es un conjunto de herramientas para automatizar los navegadores web a través de muchas plataformas que nos permitirá crear conjuntos de prueba sobre aplicaciones web. Permite la creación de scripts mediante una gran variedad de lenguajes de programación como Java, Ruby , Python, etc. [WEB44].

QTP-Quick Test Professional (pago por licencia): es el módulo de automatización de la empresa HP. Permite la automatización de casos de prueba, los scripts son programados en Visual Basic Script [WEB36].

SoapUI (libre/pago por licencia): permite probar, simular y generar código de servicios web partiendo del contrato de los mismos en formato WSDL y con vínculo SOAP sobre http [WEB40].

Cucumber (libre): permite la automatización de la prueba de aceptación para aplicaciones web. Está basada en BDD (Behaviour Driven development) y el lenguaje de generación de scripts es Ruby [WEB41].

IBM Rational Automation Framework (pago por licencia): es el módulo para la generación de scripts y automatización de pruebas de la empresa IBM que se puede integrar con los módulos de gestión. Permite la generación de scripts con varios lenguajes de programación [WEB36].