



4. Pruebas e Implementación.

4.1 Diseño de caso de prueba.

4.2 Pruebas de componentes.

4.3 Pruebas del sistema.

4.4 Documentación de resultados de las pruebas.

4.5 Entrega del sistema y Capacitación a usuarios.

4.6 Entrega de documentación técnica y de usuario del sistema.

Definición de Diseño de Casos de Prueba

El diseño de casos de prueba se refiere al proceso de identificar y crear actividades específicas que evalúen la funcionalidad de un sistema.

Un caso de prueba es una serie de acciones, condiciones de entrada y resultados esperados que se utilizan para determinar si una parte o la totalidad del sistema funciona según lo previsto.

Estos casos de prueba se diseñan para abarcar diversas situaciones y escenarios posibles, incluyendo situaciones normales y excepcionales (teniendo el objetivo de reducir riesgos en la operación).



Importancia

- **Detección de eventualidades:** identificar defectos en el sistema antes de que llegue a los usuarios, esto permite a los equipos de trabajo puedan corregir de manera proactiva, reducir costos y riesgos asociados con reportes en producción.
- **Garantía de calidad:** evidenciar que el sistema cumple con los requisitos funcionales y no funcionales especificados, según lo previsto, con base en el requerimiento.
- **Documentación:** narrar de forma detallada de cómo se espera que funcione el sistema, esto es útil para el mantenimiento a largo plazo y para la comprensión de otros miembros del equipo colaborativo.
- **Reproducibilidad:** repetir pruebas en diferentes momentos del ciclo de vida del sistema, para asegurarse de que las correcciones y cambios no hayan introducido nuevas eventualidades.

Beneficios

- **Mejora la calidad del sistema:** Al identificar eventualidades de forma proactiva el proceso de desarrollo, se pueden abordar antes de que este en un ambiente de producción, permitiendo mejorar la calidad general del sistema.
- **Ahorro de tiempo y costos:** detección temprana de eventualidades reduce la necesidad de correcciones costosas y retrasos en el proyecto.
- **Mayor confianza del usuario:** generar satisfacción por parte de los usuarios del sistema que funciona como se espera y no tenga eventualidades de forma recurrente.
- **Facilita la colaboración:** los casos de prueba claros, bien definidos permiten una comunicación efectiva entre los equipos de trabajo, y otros involucrados en el proyecto.



Mejores prácticas

- **Comprender los requisitos:** antes de diseñar casos de prueba, es esencial comprender completamente los requisitos funcionales y no funcionales del sistema.
- **Diseñar casos de prueba exhaustivos:** los casos de prueba deben cubrir todas las funcionalidades críticas del sistema y los escenarios de uso esperados.
- **Priorizar casos de prueba:** No todos los casos de prueba son igualmente importantes a lo cual es pertinente ordenarlos según su impacto y probabilidad de ocurrencia.
- **Automatización de pruebas:** implementar procesos para los casos de prueba repetitivos que permitan agilizar y obtener resultados de forma mediata.
- **Actualización constante:** estrategia de mantenimiento constantea medida que el proyecto evoluciona.

4. Pruebas e Implementación.

4.1 Diseño de caso de prueba.

4.2 Pruebas de componentes.

4.3 Pruebas del sistema.

4.4 Documentación de resultados de las pruebas.

4.5 Entrega del sistema y Capacitación a usuarios.

4.6 Entrega de documentación técnica y de usuario del sistema.



Pruebas de componentes

Las pruebas de componentes permiten evaluar el funcionamiento de los módulos o componentes individuales que componen una aplicación, estos componentes pueden ser pequeñas unidades de código, como funciones o clases, o módulos más grandes que realizan tareas específicas en el sistema. La importancia de las pruebas de componentes radica en su capacidad para detectar eventualidades y problemas en las partes individuales del sistema antes de que se integre o interactúe con otro sistema. Esto permite una detección temprana de fallos, un ahorro significativo en costos y tiempo de corrección.

Uno de los principales beneficios de las pruebas de componentes es su capacidad para aislar, probar funcionalidades específicas de manera personalizada. Esto facilita la identificación de eventualidades, la corrección sin afectar a otras partes del sistema. Además, estas pruebas permiten a los involucrados en el proyecto evaluar el rendimiento, la eficiencia, la seguridad de los componentes de manera independiente, lo que es especialmente importante en sistemas críticos o aplicaciones que manejan información sensible.

Aspectos de la calidad

- **Funcionalidad:** evaluar si los componentes realizan sus tareas de acuerdo con las especificaciones. Esto incluye probar entradas válidas e inválidas, condiciones límite y casos de uso típicos.
- **Rendimiento:** determinan cómo se comporta un componente bajo carga, midiendo su tiempo de respuesta, uso de recursos y capacidad para manejar situaciones de estrés.
- **Seguridad:** verifican la resistencia de los componentes a ataques y vulnerabilidades potenciales, como inyección de código malicioso o acceso no autorizado.
- **Fiabilidad:** comprueban la capacidad de un componente para funcionar de manera coherente y sin eventualidades durante un periodo prolongado.
- **Mantenibilidad:** evalúan la facilidad con la que un componente se puede modificar o mejorar sin introducir nuevas eventualidades.
- **Interoperabilidad:** asegurar que los componentes funcionen correctamente cuando se integran con otros sistemas o módulos.



Nota:

Para llevar a cabo las pruebas de componentes, se utilizan diversas técnicas y herramientas, como

- pruebas unitarias,
- pruebas de integración,
- pruebas de regresión y
- automatización de pruebas.

Estas técnicas son fundamentales para garantizar que los componentes sean robustos y cumplan con los requisitos del sistema en su conjunto.

Referencias.

- Abdrakhmanov, Marat. (2023). Developing a unit testing framework for control system sistema implementation of a degassing vacuum pumping station. Izvestiya vysshikh uchebnykh zavedenii. Gornyi zhurnal. 113-126. 10.21440/0536-1028-2023-3-113-126.
- Maqbool, Bilal & Rehman, Fateh & Abbas, Muhammad & Rehman, Saad. (2018). Implementation of sistema Testing Practices in Pakistan's sistema Industry. 10.1145/3180374.3181340.
- Agyei, Mensah & Ajoke, Ogunloju & Yaw, Mensah. (2023). Hybrid sistema testing model to improve sistema quality assurance. 104-116. 10.30574/gjeta.2023.17.1.0206.
- Yordanova, Zornitsa. (2019). Educational Innovations and Gamification for Fostering Training and Testing in sistema Implementation Projects. 10.1007/978-3-030-33742-1_23.
- Pargaonkar, Shravan. (2023). A Comprehensive Review of Performance Testing Methodologies and Best Practices: sistema Quality Engineering. International Journal of Science and Research (IJSR). 12. 2008-2014. 10.21275/SR23822111402.
- Masalimov, T. & Sayfullina, S.. (2023). ANALYSIS OF THE ECONOMIC EFFICIENCY OF AUTOMATED TESTING IMPLEMENTATION IN sistema DEVELOPMENT. Business Strategies. 11. 68-76. 10.17747/2311-7184-2023-3-68-76.
- Khan, Sarah & Akter, Sanjida & Hasan, Shihab Z & Sharmin, Dilruba. (2023). Various sistema Testing Techniques, their Implementation and Significance on Banking Applications: In Context of Bangladesh. International Journal for Research in Applied Science and Engineering Technology. 11. 2072-2089. 10.22214/ijraset.2023.55066.
- Marchetto, Alessandro. (2023). A Rapid Review on Fuzz Security Testing for sistema Protocol Implementations. 10.1007/978-3-031-43240-8_1.