

¿QUE ES PROBAR?

Probar es un proceso esencial para evaluar y mejorar la calidad del software, reduciendo el riesgo de fallos. Implica descubrir defectos y evaluar la calidad de los artefactos de software. Incluye pruebas dinámicas y estáticas, requiriendo planificación y habilidades analíticas por parte de los probadores.

¿PORQUE ES NECESARIO PROBAR?

Las pruebas son esenciales para el control de calidad y para alcanzar los objetivos acordados en términos de alcance, tiempo, calidad y presupuesto. Su importancia va más allá del equipo de prueba, ya que cualquier parte interesada puede utilizarlas para contribuir al éxito del proyecto. Identifican defectos en el software, abarcando componentes, sistemas y documentación asociada.

**Las pruebas
pueden ser**

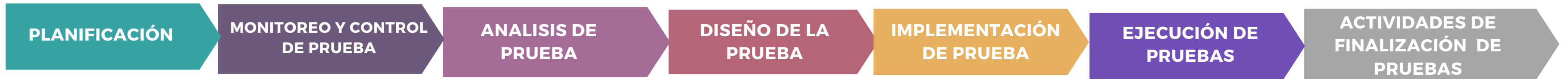
Dinámicas

Implican la ejecución de software

Estáticas

No implican la ejecución de software.

Actividades y Tareas de la Prueba



3. Las pruebas tempranas ahorran tiempo y dinero:

Detectar y corregir defectos en etapas tempranas del desarrollo ahorra tiempo y dinero, evitando la propagación de errores.

4. Los defectos se agrupan:

La mayoría de los defectos se concentran en un pequeño número de componentes del sistema, lo que permite focalizar los esfuerzos de prueba.

5. Las pruebas se desgastan:

Las pruebas repetidas pueden volverse menos efectivas con el tiempo, por lo que es necesario ajustarlas o incluso crear nuevas pruebas.

2. Las pruebas exhaustivas son imposibles:

Es prácticamente imposible probar todos los casos posibles, por lo que es necesario centrarse en técnicas de prueba efectivas y prioritarias.

1.3. PRINCIPIOS DE LA PRUEBA

1. Las pruebas muestran la presencia, no la ausencia de defectos:

Las pruebas pueden demostrar la presencia de defectos en el software, pero no pueden garantizar su ausencia

6. Las pruebas dependen del contexto:

No existe un enfoque universalmente aplicable a las pruebas; estas deben adaptarse al contexto específico del proyecto.

7. Falacia de la ausencia de defectos:

Aun con una verificación exhaustiva, no se puede garantizar que un sistema satisfaga las necesidades y expectativas de los usuarios o del cliente. Además de la verificación, la validación también es crucial.

NIVELES DE PRUEBA

- De componentes
- De integración de componentes
- Del sistema
- De integración de sistemas
- De aceptación

TIPOS DE PRUEBAS

FUNCIONALES

evalúan las funciones que un componente o sistema debe realizar. Las funciones son “lo que” debe hacer el objeto de prueba. El objetivo principal de las pruebas funcionales es comprobar la completitud funcional, la corrección funcional y la pertinencia funcional.

NO FUNCIONALES

evalúan atributos distintos de las características funcionales de un componente o sistema. Las pruebas no funcionales son las pruebas de "qué tan bien se comporta el sistema". El objetivo principal de las pruebas no funcionales es comprobar las características de calidad del software no funcional.

- Eficiencia del rendimiento
 - Compatibilidad
 - Usabilidad
 - Fiabilidad
 - Seguridad
 - Mantenibilidad
 - Portabilidad
- características de calidad del software no funcional.

DE CAJA NEGRA

se basa en especificaciones y deriva pruebas de documentación que es externa al objeto de prueba. El objetivo principal de las pruebas de caja negra es comprobar el comportamiento del sistema contra sus especificaciones.

DE CAJA BLANCA

se basan en la estructura y derivan pruebas de la implementación del sistema o la estructura interna (por ejemplo, código, arquitectura, flujos de trabajo y flujos de datos). El objetivo principal de las pruebas de caja blanca es cubrir la estructura subyacente mediante las pruebas al nivel aceptable.