Guía de estudio De ingeniería de pruebas

**Quien es el padre de la informática**

Charles Babbage

**Tipos de prueba**

Funcionales: Pruebas que verifican que el software cumple con los requisitos funcionales especificados.

No funcionales: Pruebas que evalúan aspectos como el rendimiento, la usabilidad, la escalabilidad, etc.

Prueba de rendimiento: Evalúa cómo se comporta el software bajo ciertas condiciones de carga.

Prueba de escalabilidad: es un tipo de prueba de software que evalúa cómo un sistema, aplicación o infraestructura responde al aumentar la carga de trabajo.

Prueba de integración: Prueba la interacción entre diferentes módulos o componentes del software.

Prueba Unitaria: Prueba individual de cada componente o unidad de código.

Prueba de regresión: Verifica que los cambios recientes no hayan introducido nuevos errores en el software.

Prueba de estrés: Evalúa el comportamiento del software bajo condiciones extremas.

Prueba de usabilidad: Evalúa la facilidad de uso del software desde la perspectiva del usuario final.

**Tipo de software**

* **Aplicación**: Software diseñado para realizar tareas específicas para el usuario (ej: navegadores, Word, Excel).
* **De gestión**: Software que ayuda a gestionar recursos y procesos empresariales (ej: ERP, CRM, facturación).
* **Programación**: Herramientas que ayudan a los desarrolladores a crear software (ej: IDEs, compiladores, html,js,c, java).
* **Sistema**: Software que gestiona el hardware y proporciona servicios a otras aplicaciones (ej: sistemas operativos).

**Como se mide el software**

* **Bytes que ocupa en disco**: Tamaño del software en términos de almacenamiento.
* **Número de líneas de código (LOC)**: Cantidad de código fuente que compone el software.
* **Funcionalidad que proporciona**: Capacidades y características que ofrece el software.

**Métrica para medir la calidad del software**

* Modelo COCOMO (Constructive Cost Model): Modelo utilizado para estimar el costo, esfuerzo y tiempo requerido para desarrollar un proyecto de software.

**Costos de proyectos y costos de prueba**

* Costos de proyectos: Incluyen todos los gastos asociados con el desarrollo del software, como salarios, herramientas, infraestructura, etc.
* Costos de prueba: Incluyen los gastos relacionados con las actividades de prueba, como la creación de casos de prueba, ejecución de pruebas, y corrección de errores.

**Plan de pruebas**

* Documento que describe el enfoque, los recursos, el cronograma y las actividades de prueba. Incluye:
  + Objetivos de las pruebas.
  + Estrategia de pruebas.
  + Recursos necesarios.
  + Cronograma.
  + Criterios de entrada y salida.
* **Error: Un error humano en el código o diseño.**
* **Defecto: Una anomalía en el software que puede causar un fallo.**
* **Falla o Bug: La manifestación de un defecto cuando el software no funciona como se esperaba.**

**Ciclo de vida y sus etapas**

* Análisis y diseño: Definición de lo que el software debe hacer, Creación de la
* Implementación: Codificación y desarrollo del software.
* Pruebas: Verificación y validación del software.
* Entrega: Liberación del software al entorno de producción.
* Mantenimiento: Corrección de errores y actualizaciones posteriores al lanzamiento.

**Niveles de prueba**

* Pruebas unitarias: Pruebas de componentes individuales.
* Pruebas de integración: Pruebas de la interacción entre componentes.
* Pruebas de sistema: Pruebas del sistema completo.
* Pruebas de aceptación: Pruebas realizadas por el cliente para aceptar el software.

**Casos de prueba**

* Un caso de prueba es un conjunto de condiciones o variables bajo las cuales un tester determinará si una aplicación o sistema funciona correctamente. Incluye:
  + Entradas: Datos que se introducen en el sistema.
  + Acciones: Pasos que se deben seguir.
  + Resultados esperados: Lo que se espera que el sistema produzca.
  + Resultados reales: Lo que el sistema realmente produce.