**Planificación de las pruebas**

* **¿Por qué son necesarias?:**
  + **Para verificar que el software en desarrollo es correcto y cumple con las características impuestas por el cliente.**
* **¿Cómo se organizan?**
  + **Implementando una estrategia de pruebas, como el Modelo en Espiral. Sigue el siguiente orden:**
    - **Pruebas unitarias**
    - **Pruebas de integración**
    - **Prueba de validación**
    - **Pruebas de sistema**

**Tipos de pruebas**

* **Caja Negra:**

**Se comprueba que la aplicación devuelva los resultados acorde con las entradas que recibe.**

* **Caja Blanca:**

**Se prueba la aplicación usando la lógica (No solo comprueba el resultado sino también el proceso.**

* **Pruebas de carga:**

**Para mostrar el cuello de botella de la aplicación, se comprueba cómo se comporta el programa con un alto número concurrente de usuarios (realizando una acción específica). Hay diversos tipos:**

* **Prueba de estrés: Se comprueba cuánta carga de usuarios soporta de forma simultánea hasta que se rompe.**
* **Prueba estabilidad: Se comprueba que la página soporte la carga esperada de forma continuada**
* **Prueba de picos: Se provocan cambios drásticos en la carga.**
* **Pruebas de regresión:**

**Son cualquier tipo de prueba que intenta descubrir nuevos errores y sus causas, divergencias funcionales o carencias de funcionalidad.**

* **Pruebas funcionales:**

**Evalúa si el software cumple con sus requerimientos.**

* **Pruebas estructurales:**

**Analiza la arquitectura interna.**

**Procedimientos y casos de prueba**

* **¿Cómo se diseñan y aplican?:**

**Para diseñar y aplicar los casos de prueba se necesita asegurar que se obtiene una probabilidad aceptable de detectar los errores existentes.**

**Existen varios procedimientos para el diseño de casos de prueba:**

* **Enfoque de caja negra: Nos centramos en que el programa de los resultados esperados**
* **Enfoque caja blanca: Nos centramos en que el programa concuerde internamente con las especificaciones necesarias.**
* **Enfoque aleatorio: Se elaboran casos de prueba que prueben las entradas del programa.**

**Herramientas de depuración: ¿Qué son y cómo ayudan a detectar errores?**

Errores de compilación: cuando se desarrolla se nos puede olvidar algún “;” o cerrar algún paréntesis, para solucionar eso el propio entorno nos proporciona información del fallo y así podemos arreglarlo.

Error de lógica o bugs: estos errores no evitan que el programa pueda compilar de forma correcta, pero nos puede dar valores erróneos o hacer que el programa finalice antes de lo querido.

Para solucionar este tipo de problemas, los entornos de desarrollo tienen implementada una herramienta llamada depurador. Esta herramienta permite supervisar la ejecución de los programas para localizar y eliminar los errores lógicos.

**Validaciones: Importancia del cliente en la evaluación del software.**

En el proceso de validación el cliente interviene de manera decisiva.

La validación se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad con los requisitos.

El plan de pruebas traza el tipo de pruebas que se tienen que llevar a cabo.

Cuando se procede con cada caso, pueden darse una de las siguientes condiciones:

* Las características de funcionamiento o rendimiento están de acuerdo con las especificaciones y son aceptables.
* Se descubre una desviación de las especificaciones y se crea una lista de deficiencias.

**Normas de calidad: Principales estándares utilizados en la industria.**

Los estándares que se han venido utilizando en la fase de prueba de software son:

* Estándares BSI:
  + BS 7925-1, Pruebas de software. Parte 1. Vocabulario.
  + BS 7925-2, Pruebas de software. Parte 2. Pruebas de los componentes software.
* Estándares IEEE de pruebas de software:
  + IEEE estándar 829, Documentación de la prueba de software.
  + IEEE estándar 1008, Pruebas de unidad.
  + Otros estándares ISO / IEC 12207, 15289.
* Otros estándares sectoriales.

Estos estándares no cubren determinadas facetas de la fase de pruebas. como son la organización, el proceso y la gestión de las pruebas. Ante este problema, la industria desarrolló la norma ISO/IEC 29119