**Confiabilidad**

Es el proceso mediante e cual se llevo a cabo una investigación para proporcionar a los interesados información sobre la calidad del producto bajo prueba

(Glenford J. Myers, 2004). “La prueba de software es el proceso de ejecución de un programa con la intención de encontrar errores”.

El propósito de las pruebas de software es la detección de errores en el software.

El tester debe detectar todos los posibles errores antes de que el software sea liberado para el cliente.

Composición del software: estructura interna.

Propósito de negocio: aspecto funcional.

**Verificación. -** Demuestra que un producto cumple los requisitos especificados en los hitos predefinidos durante el ciclo de vida de desarrollo. Verifica que el software cumple con los requisitos establecidos.

**Validación. -** Comprueba que el sistema cumple con los requisitos del cliente en la finalización del ciclo de vida de desarrollo. Valida que el software satisfaga las necesidades del cliente.

**Técnicas De Pruebas**

**Pruebas exploratorias**

|  |  |
| --- | --- |
| Objetivo | Aprender, diseñar y ejecutar pruebas de forma simultánea. |
| Ventajas | * Los defectos son encontrados más rápido. * Se requiere de menos preparación y planificación. |
| Desventajas | El alcance de las pruebas depende de la experiencia y habilidades del tester. |
| Uso | Se utilizan cuando la documentación de las aplicaciones no está lista. |

**Pruebas de función**

|  |  |
| --- | --- |
| Objetivo | Probar una funcionalidad a la vez. |
| Ventajas | Fácil de implementar y ejecutar. |
| Desventajas | No garantiza que todas las funcionalidades de la aplicación operen correctamente de manera conjunta. |
| Uso | Pruebas a cambios realizados en una aplicación existente. |

**Pruebas basadas en especificación**

|  |  |
| --- | --- |
| Objetivo | Verificar que la aplicación se comporta según lo especificado en la documentación. |
| Ventajas | Fácil identificación del alcance de las pruebas. |
| Desventajas | Los requerimientos que no fueron incluidos en la documentación tampoco serán verificados. |
| Uso | * Pruebas de aplicaciones que poseen documentación. * Aplicaciones que deben contemplar muchas normativas internas o externas. |

**Pruebas basadas en riesgo**

|  |  |
| --- | --- |
| Objetivo | Encontrar defectos críticos lo antes posible. |
| Ventajas | Ayuda a optimizar la priorización de las pruebas. |
| Desventajas | La identificación de las funcionalidades críticas de la aplicación puede tornarse subjetiva. |
| Uso | Pruebas de aplicaciones que sufren cambios constantemente por lo que se hace necesario limitar el alcance de las pruebas a la verificación de las funcionalidades que se utilizan con mayor frecuencia. |

**Análisis de equivalencia**

|  |  |
| --- | --- |
| Objetivo | Reducir la cantidad de pruebas dividiendo el dominio de los datos de modo que sean elegidos para las pruebas los que sean más representativos. |
| Ventajas | Permite verificar que la aplicación se comporta correctamente ejecutando una menor cantidad de pruebas. |
| Desventajas | Se pueden obviar defectos provocados por datos que no forman parte de la muestra seleccionada. |
| Uso | * Verificar comportamiento de campos numéricos. * Verificar comportamiento de funcionalidades que utilizan muchas variables. |

**Pruebas de escenario**

|  |  |
| --- | --- |
| Objetivo | Probar situaciones que pueden presentarse en la vida real. |
| Ventajas | Permite verificar que la aplicación se comportará según lo esperado ante situaciones que se presentan en el día a día. |
| Desventajas | El abuso de esta técnica puede levarnos a plantear escenarios irrealistas. |
| Uso | Probar aplicaciones en las que deben contemplarse muchas reglas de negocio y combinaciones de datos. |

**Pruebas de regresión**

|  |  |
| --- | --- |
| Objetivo | Detectar defectos producidos por los cambios realizados a una aplicación. |
| Ventajas | Permite verificar que las funcionalidades anteriores de la aplicación continúan funcionando correctamente. |
| Desventajas | Es inviable verificar todas las funcionalidades de la aplicación cada vez que se introduzca un cambio. |
| Uso | Probar aplicaciones que han sido modificadas. |

**Pruebas de aceptación**

|  |  |
| --- | --- |
| Objetivo | Validar que la aplicación cumple los requerimientos acordados con las partes interesadas. |
| Ventajas | Permite obtener retroalimentación del usuario final sobre el comportamiento de la aplicación. |
| Desventajas | El alcance de este tipo de pruebas no es garantizado, porque depende de la experiencia del usuario final. |
| Uso | Este tipo de pruebas se deben utilizar antes de poner en producción las aplicaciones. |

**Pruebas de estrés**

|  |  |
| --- | --- |
| Objetivo | Determinar la robustez de la aplicación mediante la prueba más allá de los límites del funcionamiento normal. |
| Ventajas | Permite identificar las debilidades que presentará la aplicación cuando se encuentre en producción. |
| Desventajas | * Se detectan defectos que son muy difíciles de reproducir y corregir. * Requiere personal especializado para su ejecución. |
| Uso | En aplicaciones que van a ser utilizadas simultáneamente por muchos usuarios. |

**Pruebas de rendimiento**

|  |  |
| --- | --- |
| Objetivo | Determinar el comportamiento del sistema en condiciones normales y previstas de carga máxima. |
| Ventajas | Ayuda a identificar la capacidad máxima de funcionamiento de una aplicación, así como los cuellos de botella determinando que elemento está causando la degradación. |
| Desventajas | * Requiere personal especializado para su ejecución. * Difícil análisis de los defectos detectados. |
| Uso | En aplicaciones que van a ser utilizadas simultáneamente por muchos usuarios y que además manejaran grandes volúmenes de datos. |

**Pruebas Manuales**

Es el proceso de probar manualmente el software para defectos. Se requiere un tester para desempeñar el papel de un usuario final, y el uso de la mayoría de las características de la aplicación para garantizar un comportamiento correcto.

**Pruebas Automatizadas**

Es el uso de un software especial (independiente del software que se está probando) para controlar la ejecución de las pruebas y la comparación de los resultados reales esperados.