**Control de Calidad del Software**

1\_ EL control de la calidad del software son las técnicas y actividades de carácter operativo, estas se utilizan para satisfacer los requisitos básicos de calidad, y se centran en 2 objetivos fundamentales. El primero es mantener bajo control un proceso, y como segundo eliminar las causas de los defectos en las diferentes fases del ciclo de vida

2\_ Las pruebas de software parten luego de haber llegado una fase casi terminada del software, en estas pruebas el ingeniero crea una serie de casos de prueba en las cuales se intentará demoler el software construido, este paso suele verse como algo destructivo en lugar de constructivo.

3\_ Una “prueba exitosa” es aquella en la cual se logra detectar un defecto aun no descubierto, en cambio, un “caso de prueba bueno”, es aquel en el que hay una alta probabilidad de detectar un defecto aun no descubierto

4\_ Los principios básicos de las pruebas de software son:

1. A todas las pruebas se les debería poder hacer u seguimiento hasta los requisitos del cliente
2. Las pruebas deberían planificarse mucho antes que empiecen
3. Las pruebas deberían empezar por “lo pequeño”
4. Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente

5\_ Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Esta estrategia proporciona un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuando se deben planificar y realizar esos pasos, y cuanto esfuerzo, tiempo y recursos se van a requerir. Cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de las pruebas, y la agrupación y evaluación de los datos resultantes

6\_ Para implementar con éxito una estrategia de prueba de software se deben tener en cuenta los siguientes pasos:

1. Especificar los requisitos del producto de manera cuantificable antes que comiencen las pruebas
2. Especificar los objetivos de prueba de manera explícita
3. Desarrollar un plan de prueba que haga hincapié en la prueba de ciclo rápido
4. Construir un software robusto diseñado para probarse a sí mismo.
5. Usar revisiones de técnicas formales efectivas como filtro antes de la prueba
6. Realizar revisiones de técnicas formales para evaluar la estructura de la prueba y los propios casos de prueba
7. Desarrollar un enfoque de mejora continua al proceso de prueba.

7\_

Prueba de integración: El objetivo de esta prueba es juntar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Luego de esto se debe comprobar los criterios de validación establecidos durante el análisis de requisitos.

Prueba de validación: Esta prueba proporciona una seguridad final que el software satisface todos los requisitos funcionales, de comportamientos y de rendimiento. Una vez que el software sea validado se debe combinar con otros elementos del sistema.

Prueba del sistema: En esta prueba se verifica que cada elemento se ajuste de forma adecuada y que se alcance la funcionalidad y el rendimiento del sistema total. Además, esta prueba esta construida por una serie de pruebas las cuales básicamente se encargan de verificar que todo se haya integrado adecuadamente todos los elementos del sistema y que realicen las funciones apropiadas.

Prueba de regresión: Esta prueba se encarga de volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que no se hayan propagado efectos colaterales.

Prueba alfa: Esta prueba la realiza el cliente en el lugar del área del desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y los problemas de uso.

Prueba beta: La prueba beta es una aplicación en vivo del desarrollo del software en su entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Como resultado de los problemas informados el desarrollador de software realiza modificaciones y prepara una versión n del producto del software para toda clase de clientes

Prueba de caja negra: Este tipo de prueba examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. Los métodos de prueba de la caja negra se centran en los requisitos funcionales de software. La prueba de la caja negra permite al Ingeniero del Software obtener conjuntos de condicionales de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

Prueba de caja blanca: Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Para este tipo de prueba, se deben definir todos los caminos lógicos y desarrollar casos de prueba que ejerciten la lógica del programa.

8\_ Una buena prueba tiene los siguientes atributos:

1. Una alta probabilidad de encontrar el error
2. No debe ser redundante
3. Debería ser la mejor de la cosecha
4. No debería ser ni demasiado sencilla ni demasiado compleja