# Introducción a la calidad y pruebas de software Nombre: Jose Adrian Ontiveros Moran

1. **Explica los principios de “Statical Quality Control”**  
   R: Es la aplicación de herramientas estadísticas para monitorizar la variación de un proceso o método productivo, detectar desviaciones y anticipar así posibles problemas de calidad. Se trata de una estrategia proactiva y preventiva de control de la calidad, a través de la cual las empresas pueden garantizar la estabilidad de sus procesos industriales y cumplir con las especificaciones de sus clientes.
2. **Explica el concepto de “Lean Principles”**  
   R: 1. Definir el valor=El valor es lo que el cliente está dispuesto a pagar. Es primordial descubrir las necesidades reales o latentes del cliente. A veces los clientes pueden no saber lo que quieren o son incapaces de articularlo.  
   2. Mapear el flujo de valor= El segundo principio de Lean consiste en identificar y mapear el flujo de valor. En este paso, el objetivo es utilizar el valor del cliente como punto de referencia e identificar todas las actividades que contribuyen a estos valores.  
   3. Crear flujo= Después de eliminar los residuos del flujo de valor, la siguiente acción consiste en garantizar que el flujo de los pasos restantes se desarrolle sin interrupciones ni retrasos.  
   4. Establecer el pull= El inventario se considera uno de los mayores desperdicios en cualquier sistema de producción.  
   5. Perseguir la perfección= Los despilfarros se evitan mediante la realización de los cuatro
3. ¿Qué es el diagrama de “Ishikawa”? Menciona un ejemplo de cómo construirías ese diagrama  
   R: El diagrama de Ishikawa también conocido como de espina de pescado es una herramienta visual que tiene un formato de gráfico. Además, su principal función es ayudar en los análisis de organización. La mayoría de las veces se lo emplea para encontrar la causa de un problema en su raíz.   
   Ejemplo: pues por ejemplo lo podríamos usar en una fabrica para controlar los procesos, las maquinarias, el material y los trabajadores para tener una mejor organización y estar estandarizado.
4. Explica las diferencias entre validación y verificación  
   R: verificación es el proceso de comprobar que un software logra su objetivo sin errores, y la validación es el proceso de verificar si el producto de software está a la altura o, en otras palabras, el producto tiene requisitos de alto nivel.
5. Explica y menciona un ejemplo relacionado al software de failure, error and fault  
   R: software de failure= Cuando una computadora muere por un fallo de software y no por un problema de hardware, sus archivos permanecen inaccesibles pero intactos en el disco duro.  
   error and fault=es un problema en un programa de computador o sistema de software que desencadena un resultado indeseado.  
   fault: se refiere a cualquier funcionamiento incorrecto del Software Suministrado que resulte en que el Software Suministrado no funcione sustancialmente de acuerdo con las normas establecidas en la Especificación Funcional o cualquier especificación aplicable acordada entre las partes
6. ¿Cuál es la diferencia entre “fault injection” y “fault simulation”?  
   R: fault injection: Es un método de prueba de software que introduce deliberadamente errores en un sistema para asegurarse de que puede resistir y recuperarse de las condiciones de error.  
   fault simulation= evalúa cómo se comportará un circuito digital en presencia de defectos de fabricación.
7. Menciona un ejemplo de pruebas de caja blanca y pruebas de caja negra. Explica el concepto de pruebas de caja gris.  
   R: Ejemplo de caja negra un algoritmo de caja negra es uno donde el usuario no puede ver el funcionamiento interno del algoritmo.  
   Ejemplo caja blanca   
   Printme (int a, int b) { ------------ Printme is a function

int result = a+ b;

If (result> 0)

Print ("Positive", result)

Else

Print ("Negative", result)

} ----------- End of the source code

Para ejecutar las declaraciones en el ejemplo de prueba de caja blanca anterior, habría casos de prueba de WhiteBox

A = 1, B = 1

A = -1, B = -3  
Caja gris Las Pruebas de caja gris , también llamado análisis de caja gris, es una estrategia para la depuración de software en el que el tester tiene un conocimiento limitado de los detalles internos del programa. Un cuadro gris es un dispositivo, programa o sistema cuyo funcionamiento se entienden parcialmente.

1. Explica los conceptos de prueba unitaria, integración, sistema, aceptación y regresión  
   R: Prueba unitaria= Las pruebas unitarias son verificaciones de partes de código como en un método o función.  
   Prueba de integración=Estas pruebas se hacen después de que las pruebas unitarias pasaron es la integración de todos los módulos y funciones corran bien entre si.  
   Prueba de sistema: El objetivo del sistema de pruebas es encontrar un error para determinar situaciones en donde algo pasa cuando no debe de pasar y viceversa.  
   Prueba de aceptación=estas prácticas que permiten obtener los primeros comentarios de un número limitado de usuarios finales y clientes, o incluso de los clientes de éstos.  
   Prueba de regresión= el objeto de descubrir errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa.
2. Define un proyecto de software, ya sea personal u open source, después identifica fuentes para derivar casos de prueba.

