**TESTING**

**Contexto:**

El testing no es la actividad responsable de asegurar la calidad ni desarrollar software de calidad

Porque la calidad del software no depende solamente de las características del producto.

**Necesidad del testing**

Se vuelve necesario ya que partimos de que nuestro producto de software tendrá errores y sin inevitables.

JAJAJA No sirve para rellenar el tiempo

Para evitar ser demandado mi client ¿?

Para reducir riesgos

Tener confianza en mi producto

Las fallas son costosas

Para asegurarno que nuestro producto se ajusta a los requerimientos de mi cliente y que las funciones se implmentan correctamente.

#El testin es un proceso destructivo cuyo objetivo es encontrar defectos#

**Definicion**

**##**

Se asume que ecisten defectos

La actitud del testing es negativa porque mi intención no es probar que anda sino donde se rompe el sistema.

En sistemas confiables el 30 p 50 % se lo lleva el testing en términos de ocstos

**Principios**

Debe mostrar defectos

El testing exhaustivo es imposible

El objetivo es abarcar la mayor cantidad de casos posibles con el menor esfuerzo posible,

Testing temprano -> El testing comienza con la escritura de casos de pruba y lo planifico

Agrupar defectos según los casos de prueba y el universo de casos

Paradoja del pesticida:= Me enfrasco tanto en matar una plaga en particular que las demás ocupan su lugar y no me doy cuenta de ese fenómeno

En el testing ocurre lo mismo poruqe me concentro en un tipo de defecto y los demás tipos se me podrían pasar , esto es importante especialmente para el testing automáticamente.

#Se pueden dejar defectos sin atender#

El programador debe evitar su propio código

El primer nivel de testing es donde el programador puede encontrar defectos en su código.

No se debe planificar el esfuerzo de teststin en función que todo anda de maravilla.

Orientacion para encontrar edefectos no es solo en que SW haga lo que tenga que hacer pero también que no haga lo que no debería hacer ¿?(prueba por el lado positivo y la prueba por el lado negativo)

**Cuanto es necesario testear**

#Test de regresión: si la corrección no trae nuevos defectos#

If it´s is imposible, what can i do?

LO QUE SE DEBE TENER EN CUENTA

Cuando se tiene la confianza de que el software anda bien

Pero depende del riesgo del software, no es lo mismo rappi que un sistema automatico de inyección de “vacunas”

**Conclusiones**

El nivel de exposición a riesgo es una variable directa del testing.

Los costos asociados del proyecto son una variable directa del testing.

En función de los niveles de riesgo se puede establecer prioridades para los casos de prueba.

Criterio de aceptación

Es un acuerdo con el cliente hasta donde se va probar ya que me permite respaldar mis decisiones. Debe ser medible

Términos

Costos

% de test corridos sin fallas

*Fallas predichas aún permanecen en el software(Recordemos que lo que entregemos va a tener defectos)*

No hay hay defectos de una determinada severidad en el software

#Es una raro pensar que un sistema no tiene errores#

#Es necesario se claros con el cliente#

**La pscologia del testin**

La mentalidad de quien testea no es criticar la habilidad del equipo sino que existen defectos always entonces se evita la confrontación.

Es un actividad orientada a la calidad del software que se desarrolla.

**ERROR VS DEFECTO**

// Paré en el minuto 33 de la primera parte de testing