**Error vs Defecto**

Se habla de error se detecta en la etapa que se “introduce”

Se habla de defecto cuando se detecta en la etapa posterior que se “introduce”. El testing busca defectos.

**Conceptos:Defectos, severidad y prioridad**

Los defectos se clasifican en 2 variables

Severidad :La debemos tener en cuenta en los criterios de aceptación

Tiene que ver si los defectos permanecen en el software. Es en el contexto del caso de prueba.

1 Bloqueante :=No puedo seguir ejecutando el caso de prueba

2 Critico :=A nivel de funcionalidad se ve afectada, no funciona adecuadamente

3 Mayor := Lo mismo de arriba pero menos critico

4 Menor := El defecto ocurre en caminos alternativos,

5 Cosmetico := Etiquetas errores de ortografia

Prioridad: Esta relacionada a la prioridad de la funcionalidad dentro del negocio. No esta relacionada a la severidad

1 Urgencia:=

2 Alta:=

3 Media:=

4 Baja:=

# El testin debe contemplar lo funcional y lo no funcional #

**Niveles de prueba**

Tienen que ver como se va escalando lo que voyprobando de menor a mayor granularidad

1 Prueba unitaria:= Se prueba cada componente de manera individual e independiente.Lo suele hacer el programador.Suele ser el primer testing automatizado.Se utilizan herramientas de debugeo.

Se ejecuta el testing, se lo prueba y se lo corrige

2 Testin de integración := Cuando se juntan los componentes, se lo hace de manera incremental, experimentando con cada incremento de a pco.Se integra primero lo mas critico a lo menos critico

3 Testin de sistma : Se experimente al sistema como un todo teniendo en cuenta lo funcional y lo no funcional seguridad ,carga,performance y se intenta prbar en un ambiente parecido al de producción.

4 testin de aceptacion : Es el que realiza el usuario para saber si el sistema se ahusta a sus necesidades, aca el foco esta en otro lado en vez de buscar defectos sino si el sistema se comporta como el cliente desea. Aca s erealzian las priebas alfa y beta. Relacionandolo con scrum en qué ceremonia se realiza este testin?En la review ( es donde el cliente po mira el software según las feature que entregamos)

**Proceso de rueba según los procesos definidos**

Hay diferentes epatas

1 Se defina el planificación, es un parte pequeña del plan de protyecto

Se construye el plan de pruba

Se define la autmatizacion de pruebas

Los riesgosy objetivos

Se debe modificar todo el timpo

2 Identificacion y especificación de los casos de pruebaqué vamoa a probar

Determinar lo que se va aprobar

Identificar los casos de ruebas

Definir las entradas

Priorizacin de los casos de pruba

Se puede probar sin código

Deteminar las salidas esperadas

3 Se ejecutan los casos de prueba

Automatizar los casos de prueba que se puedan

Se vuelve priorizar

Se definen los casos de prueba en un sprint de preubas

Ejecutar

Comparar los resultados obetnidos con los deseados para saber si pasa o no los casos de prueba y se define su severidad y urgencia

4 Se analizan las fallas si se no se cumple con el criterio de aceptación entonces se vuelva ejecutar ( Supongo que previamente se intenta arreglar el bug, leer los apuntes de diseño)

Evaludar los criterios de aceptacion

5 Terminar las prubas

Cuantas líneas de código necesito¿ CEROJEJE

Y en scrum qué onda¿ El testin se hace embebido en el sprint, solamenteque no hay un proceso definido

Encontrar los defectos lo ante posibles y arreglarlo lo antes posible

Es una actividad mas del sprint

**Caso de prueba CP**

Es la unidad de actividades de prueba

Tiene que ver con escribir una secuencia de pasos que tiene condiciones o variable que permite saber si el software funciona o no.

Lo importante es que las variables y condiciones claras y conjunto de pasos e claro para poder realizarlo esperando el mismo resultado lo deseado o no. Para quien tenga que arreglar pueda reproducir el error, siempre de la misma manera.

Condicciones de prueba.0 Es importante para reproducir el caso de prueba

Consta de 3 parte el CP

1 Objetivos. Tiene que ver con lo qué voy a probar concretamente, la característica

2 Condicion de prueba datosa introducir al sistema que se encuentra en condiciones preestablecida

3 Comportamiento esperado.0 La salida a la acción esperada en el sistema de acuerdo a los requerimientos del mismo

Se deben tener en cuenta que como el testin exhaustivo es imposible, entonces se debe probar lo que mas se pueda con el menor esfuerzo. Intentado el máximo porcentaje de cobertura con el menor número de casos de prueba. Hay una premisa que dice que los bugs se congregan en lo limites por ende hacer casos de prueba que se oriente a probar los datos extremos. Elegir donde probablemente haya errores. Hay técnicas para probar estas cosas

COMO HAGO PARA ENUNCIAR CASO DE PRUBA

Si hay requerimieinteos planteados o user story, se puede empezar desde ahí

*MUY IMPORTANTE.0 LAS PRUEBAS DE USUARIO NO SON CASOS DE PRUEBA sirven para los criterios de aceptcion.LOS CASOS DE PRUBA NO ESTAN EN LA TARJETA DE LA USER. DESDE EL EQUIPO PARA ADENTRO. AYUDAN PERO NO DEFINEN*

Si no hay user, se usan los casos de uso

Hay técnicas que determinar los requerimientos guiados po casos de prueba. En lugar de escribir los requerimientos con forma de casos de pruba. No se indga

Si no tengo esas cosas, pueden surgir del código

Sino desde los documentos ddel cliente

Entre mas claros los requerimientos mas fácil hacer los casos de prueba

**Conclusiones sobre la generación de casos.**

Ninguna técnica es completa

Por lo que se desea usar una combinación de técnicas para intetar abarcar la mayor cantidad de software con el meno esfuerzo, poque hay difentes técnicas que me permite probae difentes cosas, Apuntan a cosas distintas.

Se debe tener bien especificado los requeriemientos es exponencial la mejora.

ABARCAR LO QUE MAS SE PUEDE CON EL MENOR NUMERO DE CASOS DE PRUEBA.

Ser sistematicosy documentra las suposiciones sobre el comportamiento

**Ciclo de test nuevo concepto**

Sea un conjunyo de casos de pruba cp, va a haber difentes resultados unos con defectos y otros no. Algunos defecto mas graves que otros. Una vez que se corrige se vuelven a ejecutar los casos de prueba y se realiza un nuevo ciclo de test. Es un conjunto de pruebas a una versión del sistema una vez que corregimos los defectos encontrados, se hace hasta que la definición de criterio de aceptación se cumpla

**Regresion**

Es cuando se termina un ciclo de prubba y se sustituye la versión por una corregida, es importante probar de vuelta todos los casos de prueba ya que pueden surgir nuevos defectos de la corrección y los que antes salían bien ahora pueden salir mal.

El punto aca identificar los nuevos errores que se introducne por la corrección de errores

**Smoke de test**

Hacer una primera corrida del sistema sin ser exhaustivo para probar que no hay errores catastróficos y corregir los errores más groseros antes de ponerse a corregir detales

**Testing funcional**

Esta basado en los requerimientos

**Testing no funcional**

Es la prueba de cómo funciona el sistema

NO HAY QUE OLVIDARSE por que los no funcionales son tan importantes como los no funcionales

Performance testing

Prueba de carga

Prueba de servicio

Performance

Prueba de mantemiento

Prueba de fiabilidad

Es importante probar el sistema en ambientes muy parecidos al de producion sino es difícil ejecutarlo

Uno se suele focalizar en lo funcional

**Prueba de interfacesde usuario**

Ahora son muy importantes tanto que merecen su análisis importante

**Prueba de performance**

**Prueba de configuración**

Probar mi entorno con todas las configuraciones posibles

Cierre de clase

Cuando se habla del testin y los ciclos de vida, ya que hay diferentes ciclos de vida

Actividades importants en los ciclos de vida

Verificacion y validación.