Testing de software PPT 13

aseguramiento de calidad de proceso (verlo por nuestra cuenta, mas adelante vamos a ver si va al parcial o no)

que es el testing? objetivo del testing?

encontrar defectos. el testing es exitoso cuando encuentra defectos, si no encuentra defectos no es exitoso. EL TESTING EXITOSO ES EL QUE ENCUENTRA DEFECTOS

siempre hay defectos, el punto esta en hasta cuando toleramos los defectos.

30-50% es el peso que tiene el testing en el desarrollo de software

la calidad del software no depende de si hacemos mucho o poco testing.

error y defecto no son lo mismo.

el error se encuentra en la misma etapa que se produce. el defecto se detecta en una etapa posterior.

es mas caro corregir defectos.

SEVERIDAD Y PRIORIDAD

es un ejemplo de severidad, nosotros podemos usar el que queramos

severidad: impacto que tiene el defecto en el uso del software

prioridad: tiene que ver con el impacto en el negocio. que tan urgente es para resolver.

niveles de prueba

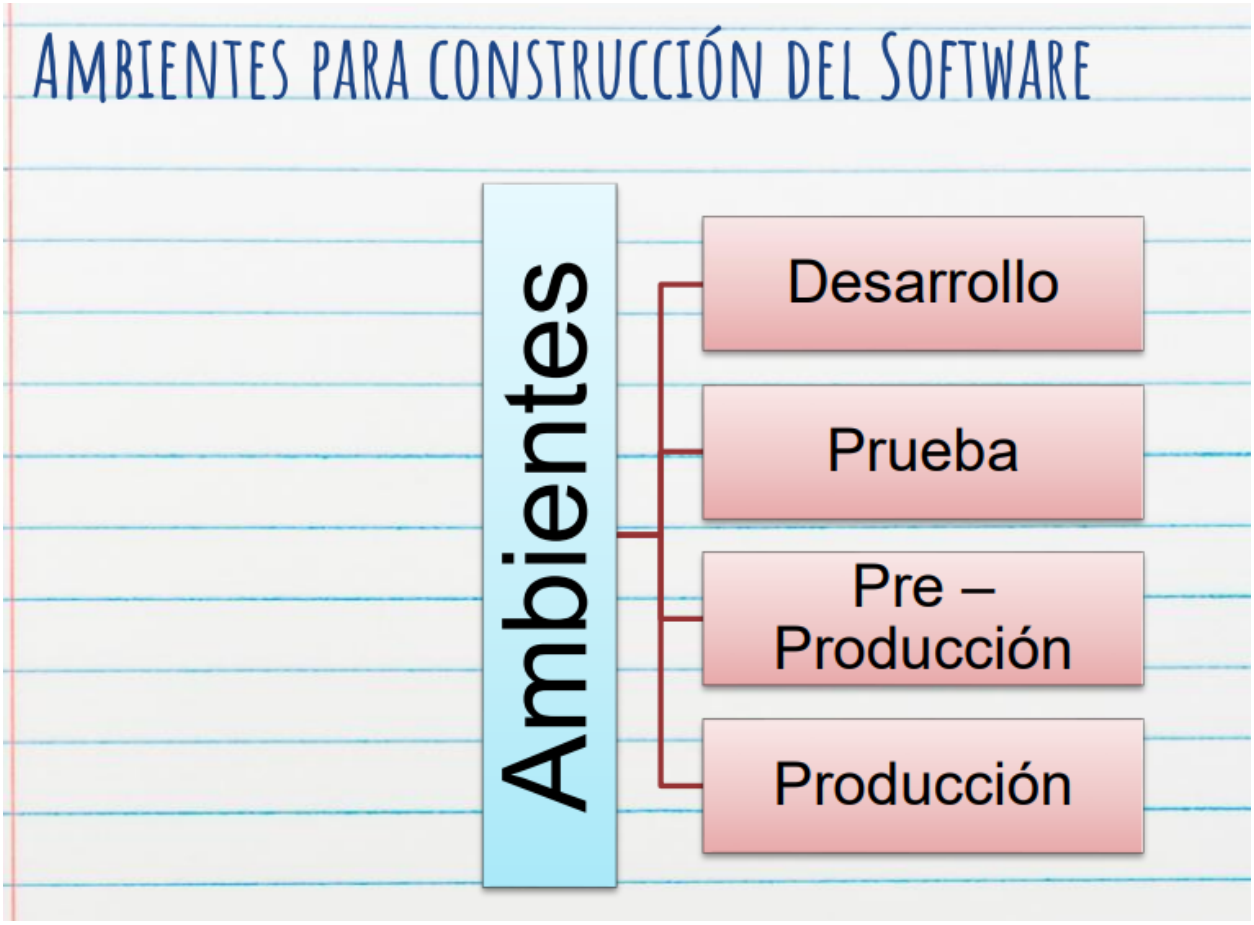
4 niveles de prueba

unitario se prueba cada componente de forma individual y forma independiente. en casi todo los casos es un testing automatizado, o es lo que se espera. aca se habla de error, porque puede ser que lo encuentre el mismo desarrollador en la misma etapa, pero en algunos lugares se suele usar como sinonimo. son probados por quien genero el codigo

integracion no es probado por quien hace el codigo, para sacar la subjetividad. puede ser automatizable en algunos casos. sirve para ver si funciona cuando integramos las partes aisladas.

sistema nos permite ver como funciona la aplicacion de forma completa, pruebas que permitan verificar requerimientos funcionales y NO FUNCIONALES. tiene que estar lo mas cercano a cuando se entrega a produccion. en las otras pruebas es mas dificil validar requerimientos no funcionales

aceptacion: su objetivo es si el software es lo que el cliente quiere, no esta para encontrar defectos. pero si se pueden encontrar.



casos de prueba: conjunto de instrucciones de como ejecutar determinada funcionalidad para ver si encontramos defectos o no. ayuda a encontrar defectos.

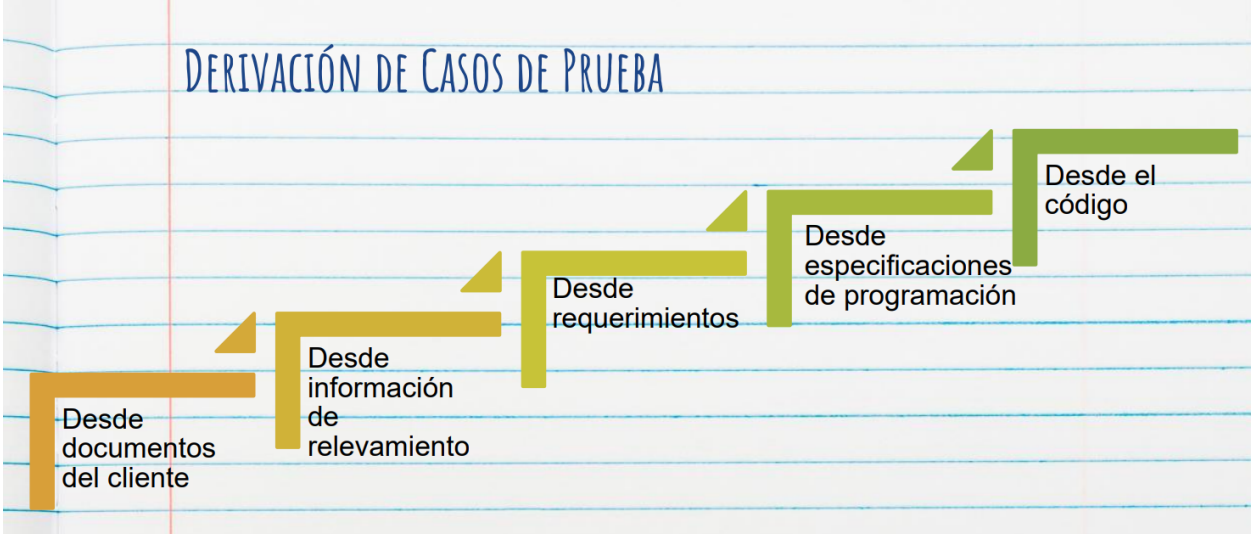
hay que escribir casos de pruebas que nos ayuden a maximizar la mayor cantidad de defectos.

enrealidad cuando escribimos los casos de pruebas, en los limites bordes o esquinas, es donde se congregan los defectos. por ejemplo al pasar un precio a cobrar, capas hay algun error en un precio cercano a ,99 o valor negativo.

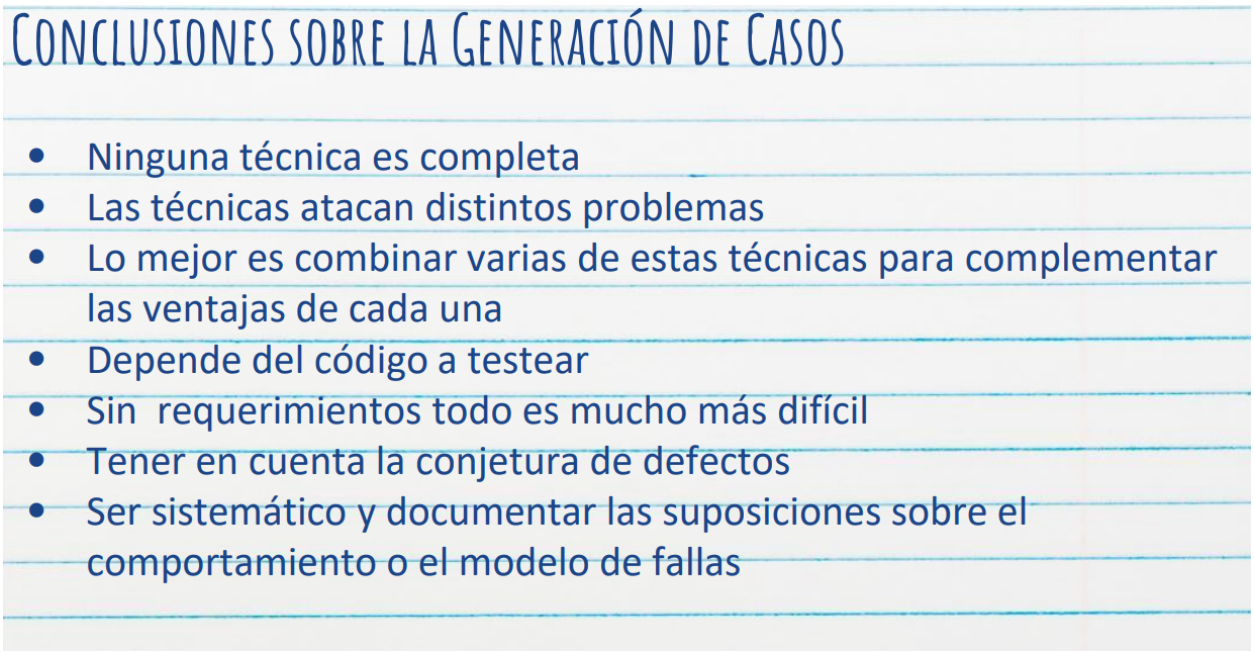
si los requerimientos estan bien especificados es mas facil encontrar los defectos.

mientras mejor especifiquemos los requerimientos es mas facil hacer el testing

Fuentes:



tips para la generacion de casos de prueba



este año no se va a trabajar el concepto de caja blanca en el PRACTICO.

condicion de prueba

Esta es la reacción esperada de un sistema frente a un estímulo particular, este estímulo está constituido por las distintas entradas.

casos de prueba

un caso de prueba es la unidad de la activvidad de la prueba.

no se usa `se ingresa el numero de cliente` sino que se usa `se ingresa el numero de cliente 123`

hay que usar un ejemplo concreto.

caja blanca y caja negra

suelen ser complementarios, no suele ser uno o otro.

en ambas necesitas escribir casos de prueba, sin caso de prueba no puedo probar

CAJA NEGRA

- basado en especificaciones

  es el metodo en el cual ingreso ciertas entradas con ciertas condiciones, y ver si las salidas coinciden con lo esperado. hay que tratar de escribir el menor numero de casos de prueba para encontrar la mayor cantidad de defectos.

- basado en la experiencia

  no lo vamos a ver desde el practico, adivinanza de defectos.

CAJA BLANCA

- estructura interna del sistema, logras toda la cobertura, ves todo lo que pasa dentro de software.

- combinando las entradas.

CICLO DE TEST

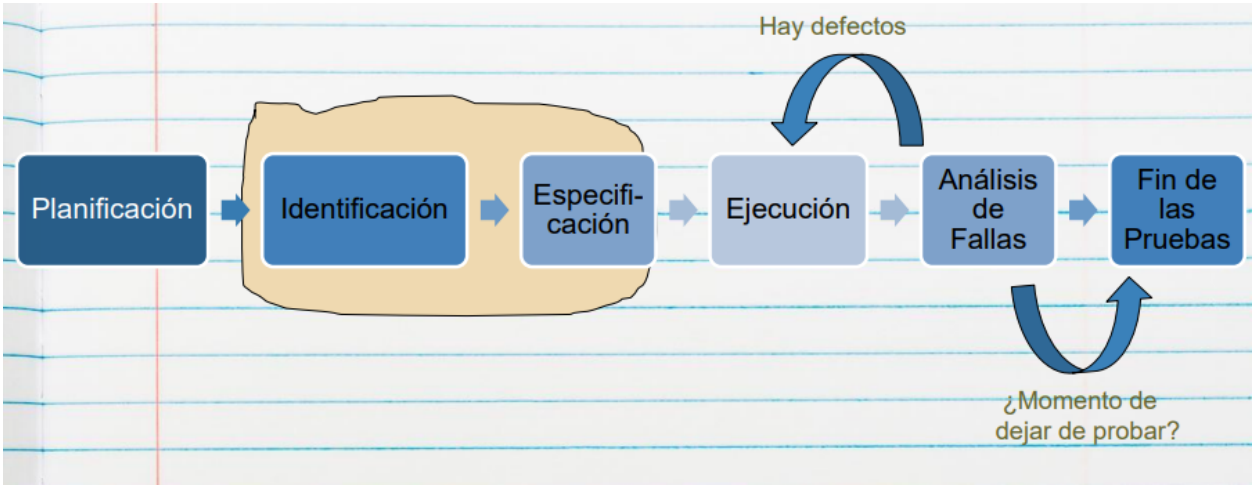
conjunto de casos de pruebas, una ejecucion de cada caso de prueba, termina el ciclo de testing.

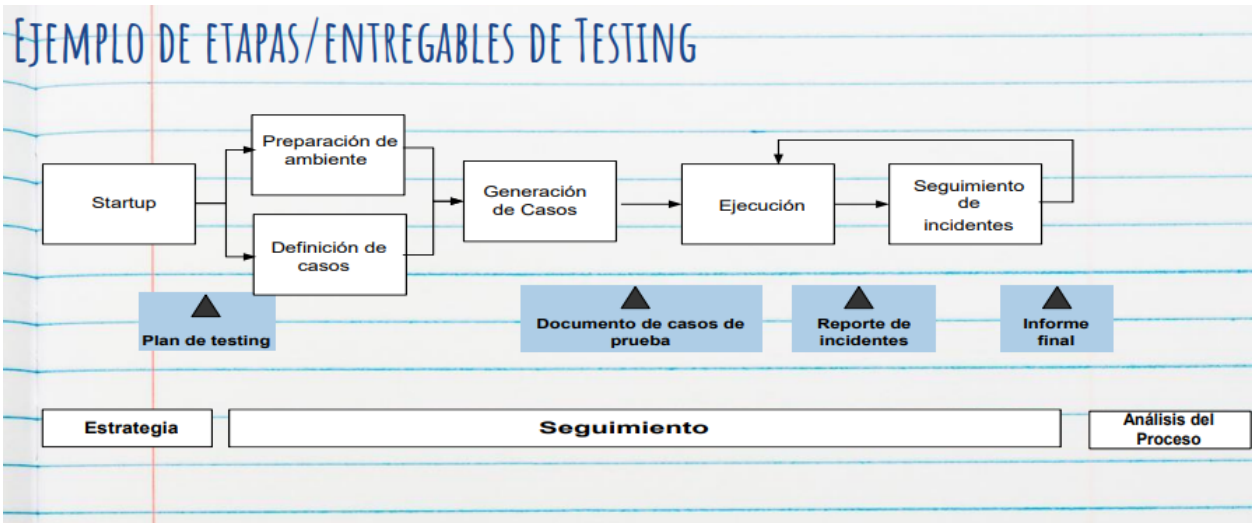
TEST DE REGRESION

si termine el ciclo de prueba, corregi, probablemente no tenga que ejecutar los casos de prueba que me tiro defectos, sino que tengo que probar todo, porque asumo que cuando corrijo un defecto puede aparecer otro

Proceso de pruebas

EJEMPLO:





con el cliente definimos cuando termina el ciclo de testing.

PROCESO DE TESTING

planificacion y control

indentificacion y especificacion

ejecucion

evaluacion y reporte, tiene que ver con ver si cumplieron las pruebas de aceptacion.

EL SOFTWARE NO SE PUEDE PROBAR DE MANERA COMPLETA

EL TESTING EXHAUSTIVO ES IMPOSIBLE

Hay que definir el criterio de aceptacion. Se usa para resolver el problema de determinar cuando una determinada fase de testing ha sido completada, puede ser en termino de costos, porcentaje de tests corridos sin fallas, etc.

Fuera de los test unitarios, nadie deberia probar su propio codigo.

no hay que suponer que no se va a encontrar defectos.

SMOKE TEST

primer corrida de los test de sistema, que provee cierto aseguramiento de que el software que esta siendo provvado no provoca un fallo que no permita ni loguerase por ejemplo.

TIPOS DE PRUEBA

Testing funcional: tenemos determinada caracteristica y probar si dicha caracteristica es lo que yo espero

Testing no funcional: tiene que ver con los rnf.

TDD

es empezar el desarrollo haciendo los test, sin escribir ninguna linea de codigo.

es escribir codigo hasta que el test dé correctamente.