Diapositiva 3:

El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante laspruebas beta, es posible hacer modificaciones y luego preparar la liberación del producto desoftware a toda la base de clientes

**Pruebas unitarias**

Saber cómo tiene que reaccionar el programa ante distintas circunstancias

**Valores frontera:**

Son los valores límite, ya sean máximo o mínimos que puede recibir un programa y que nos ayudan a verificar su comportamiento

Estas pruebas las realiza el desarrollador, y consta de tres etapas:

1.- preparación (diseño)

2.- ejecución

3.- la comprobación(revisar si devuelve los datos esperados)

**Pruebas de integración**

Probar cada función por separado, se utilizan para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz

*Integración incremental*

La integración incremental es cuando el programa se construye y prueba en partes

*Integracion ascendente*

Comienza la construcción y pruebas con modulos atomicos, se agrupan componentes, y este se convierte en uno nuevo y el mismo proceso se realiza de abajo hacia arriba

**Pruebas de regresión**

**Se aplica para asegurar que no ocurran errores colaterales, intenta descubrir bugs, es decir el programa hace algo que no debería, se aplican cada vez que hay un control de cambios**

**Pruebas de humo**

Es un planteamiento de prueba de integración que se usa cuando se desarrolla software de producto, es decir es un software que no se entrega por etapas, cuando los tiempos de entrega son más cortos de los que puedan ser normalmente, se hace para estar probando de manera constante, para no esperarse al final a probarlo, si no sobre la marcha

**Actividades que abarca**

1.-Los componentes de software traducidos en código se integran en una construcción

2.-Se diseña una serie de pruebas para exponer los errores

3.-Las construcción se integra con otras construcciones

**Beneficios**

1.-Se minimiza el riesgo de integración

2.-La calidad del producto final mejora

3.-El diagnóstico y la corrección de errores se simplifican

4.- el progreso es mas fácil de valores

**Pruebas de validación**

*Se realizan después de las pruebas de integración, tienen como base los requerimientos del cliente, se podrían considerar también la documentación solicitada y no solo la funcionalidad del producto*

Proceso en el cual el desarrollador verifica si cumple con lo que el cliente pude, nos pide para determinar los requerimientos que el cliente propuso desde un principio

Va después de integración para verificar si el producto se construyo correctamente, esta bien si satisface las necesidades del cliente.

Se dedica mas al funcionamiento del producto

Busca que satisfaga los requerimientos en todo momento

**Pruebas del sistema y recuperación**

Integra diferentes pruebas con el propósito de ejercitar por completo el sistema. Cada prueba tiene un propósito diferente

**Lo que abarca:**

* **Pruebas de recuperación**
* **Pruebas de Seguridad**
* **Pruebas de Esfuerzo**
* **Pruebas de Rendimiento**

Sus propósito es el manejo de errores que prueben toda la información proveniente de otros elementos del sistema

Validar la información que se estará manejando en el sistema, simular los errores que pueden ocurrir, registrar los errores o evidencias

**Dedo acusador: se culpan unos a otros del problema que detectaron**

* **Pruebas recuperación**

Verificar que los procesos de recuperación respalden la base de datos

***Condiciones que deben incluirse***

* **Interrupción de electricidad en el cliente (luz en oficina) y/o servidor**
* **Interrupción de comunicación hacia el servidor**

La base de datos, aplicaciones Deben retornar a un estado conocido y deseado

**Consideraciones especiales**

**Quien va a estar presente cuando ocurra el fallo(desarroladores,operadores)**

Seguridad

Son aquellas que intentan verificar que lops mecanismos de vverificacion realmenta protejan al sistema de los ataques

Debe proarse para ser vulnerables ataques frotnales: por donde entran los datos

Laterales: entran por medio de cosas aejnos al programa

Puerta trasera: por donde solo pueden salir datos

Trata de romper el sistema para ver los puntos débiles, estas pruebas osn sugeribles en hacerlas en cada fase del ciclo de vida

Papele diseñador sistema

Cuando alguein ataquie el software obtenga la menorinformacion posibles y que sea mas dificl obtener esta info

Surgen vulnerabiidades por un mal diseño, deficiente implementación del sistema informatico. Las normas obligan a hacer pruebas

Caja gris: se le da info

Caja blanca: se le cierta da info

Caja negra: no se le da info

Pruebas de esfuerzo

Procedimiento que te ayuda a saber que tanto puede soportar tu software, lleva al software a un limite en como será en la vida real,se basa en realizar pruebas en casos anormales

//Pruebas de esfuerzo (¿Cuánto podemos doblar esto antes de que se rompa?).

Se ejecutan en un sistema en forma que demanda recursos en cantidad o frecuencia anormal. Existe una variación llamada prueba de sensibilidad que intenta descubrir combinaciones de datos que puedan causar inestabilidad o procesamiento inadecuado

Pruebas de rendimiento

Son para diferentes propósitos, de aquí se deslindan otraspruebas que se van haciendo en bloques

Sistema: objeto sobre el que se hará la prueba, puede ser aplicación software, componente o plataforma hardware

Carga: indicador de concurrencia sobre el sistema

Concurrencia: que tan seguido recibe información y procesándola

Tipos:

Pruebas de carga: intentan alcanzar el objetivo principal del software

Pruebas de capacidad: objetivo encontrar el límite, hasta donde va a alcanzar ese software

Pruebas de estrés:

Someter al sistema con mas carga

Pruebas de estabilidad: comprueba que no exista un degradación

Pruebas de auskanuebti:pruebas por componentess, es decir los aíslan de los demás

Uno de sus problemas mas comunes es que se tiene que determinar el momento en el que se van a realizar

Pruebas de despliegue

Se llaman pruebas de

Este tipo tiene un objetivo, probar el producto en su entorno de ejecución final, empaquetar el software para su distribución. El punto ´rincipal del despliege es que los clientes sean independientes

Verificación y validación del software en las estaciones de trabajo de los usuarios, requerimientos mínimos

Aquí se genera el manual de usuario,cuando s egenera usuario en diferentes estaciones de trabajo se tiene que hacer una representación correcta para ver que el funcionamiento sea