08/10/2020

PROCESO GENERICO DE TESTING

INGENIERO. ABEL SOSA ESCOBEDO

TECNOLOGIAS DE LA INFORMACION AREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA

EQUIPO DINAMITA

ANDRES SALVADOR RODRIGUEZ MEJIA | 19170163
MARCOS AMAURY RODRIGUEZ RUIZ | 19170087
LUIS FERNANDO MARTINEZ MOYA | 18090022

4°B

Proceso genérico de Testing

Estandar ISO 29119 para pruebas de software.

Configuracion del ambiente de pruebas, investigación y entrenamiento.

Diseño y ejecución de casos de prueba.

Admnistracion de defectos, Regresion de pruebas (release) y sign off, Validacion de deployment.

El dia a dia de un tester y los mecanimos de comunicación con otros roles.

Proceso genérico de Testing

¿Qué es el testing de software?

El Testing de Software es la realización de pruebas sobre el mismo, con el fin de obtener información acerca de su calidad.

Objetivos

- Encontrar defectos o bugs con el finde ser subsanados.

- Eso nos va a permitir aumentar la confianza en el nivel de calidad del software. Aunque por muchas pruebas que realicemos no vamos a poder afirmar al 100% que ese software no contiene errores.

- Esta calidad nos va a facilitar la toma de decisiones. Por ejemplo, si tenemos una entrega que debe pasar a producción, si tiene muchos defectos y la calidad no es buena, a lo mejor debemos posponerlo y subsanar primero los errores, pero si la calidad es buena y no tiene errores, lo pasamos a producción.

- Evitar la aparición de nuevos efectos, ya que si aumento la calidad de mi software y hacemos las cosas bien, es menos probable que aparezcan nuevos defectos que si hago las cosas de cualquier forma y con una calidad baja.

¿Qué evitamos aplicando las herramientas y técnicas para asegurar la calidad del software?

- Retrasos en la entrega acordada con el cliente.
- Descontento del cliente por los problemas en el desarrollo del producto digital.
- Sobrecoste económico. Nuestra experiencia nos demuestra que de forma cuantitativa y cualitativa los costes para mantener la calidad del software son inferiores a los que se producen por la falta de aseguramiento de la calidad.
- Descontento por parte de los usuarios. Que buscarán rápidamente otra opción.

Tipos de pruebas de Testing

Según si ejecutamos o no el código

Las estáticas son aquellas que se hacen sin necesidad de ejecutar el código. Un ejemplo de este tipo de pruebas puede ser la revisión estática de código, es decir, analizar el código fuente de una aplicación en busca de defectos, de algún tipo de patrones incorrectos y demás.

Las pruebas dinámicas, en cambio, son aquellas en las cuales tengo que ejecutar el software para poder probarlo.

Según el uso de herramientas

Las pruebas manuales son aquellas en las que se prueba una navegación normal, por ejemplo, o se realiza una prueba funcional, como por ejemplo, acceder a la aplicación y pulsar los botones para comprobar si funciona o no.

Las pruebas automáticas se usa una herramienta para realizar estas pruebas, por ejemplo una prueba automatizada, en la que grabo una navegación y luego ejecuto esa prueba de forma automática desde la herramienta.

Las pruebas funcionales, que a su vez existen varios tipos, de las que las más importantes son las unitarias, de integración, de aceptación y de regresión.

- **Las unitarias** son pruebas para trozos de código concretos para ver que funciona y que no tiene errores.

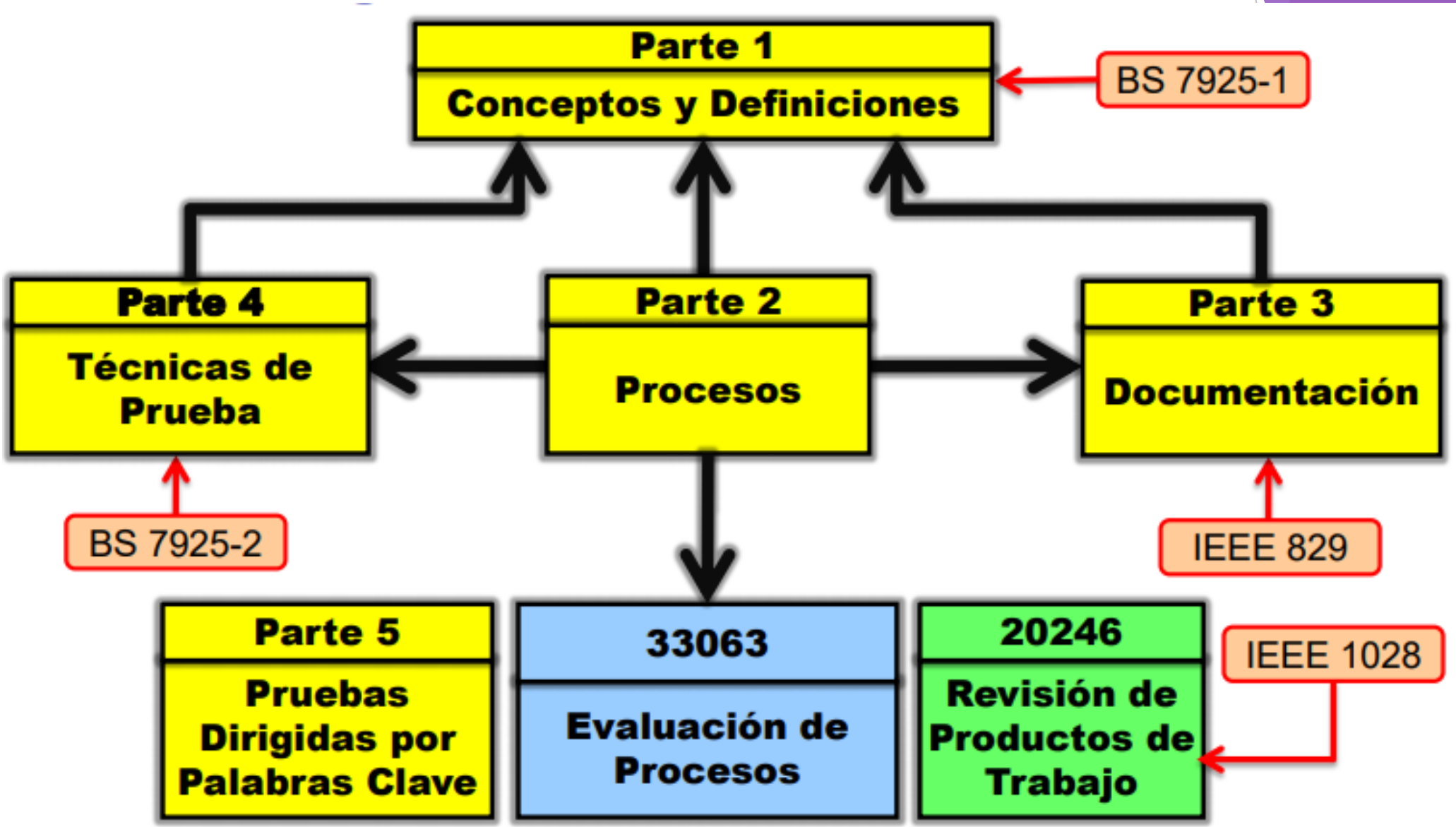
- **Las de integración**, que son pruebas a todos los componentes juntos, para ver cómo interactúan entre ellos y comprobar que todo vaya bien.

- **Las de aceptación**, que las suele hacer el usuario. Puede que un software no contenga errores, que funcione bien, pero tal vez no hace lo que debería hacer, no está haciendo lo que el usuario quería que hiciese.

Las pruebas no funcionales, que por el contrario son aquellas que no se basan en aspecto funciona.

- **De seguridad**, el famoso hacking ético, en las que se buscan vulnerabilidades de seguridad.
- **De rendimiento**, que permiten conocer el comportamiento del software ante una carga determinada, cómo responde y cómo se recupera ante fallos.
- **De usabilidad**, que se emplean para saber cómo de usable es la aplicación, pero sin entrar en aspectos funcionales. Por ejemplo, si tiene un menú que hace que la navegación sea intuitiva, si tiene una ayuda que explica el funcionamiento de la aplicación, etc.

Estándar ISO 29119


Parte 1
Conceptos y Definiciones
BS 7925-1
Parte 4
Técnicas de Prueba
Parte 2
Procesos
Parte 3
Documentación
BS 7925-2
IEEE 829
Parte 5
Pruebas Dirigidas por Palabras Clave
33063
Evaluación de Procesos
20246
Revisión de Productos de Trabajo
IEEE 1028

Configuración de un entorno de pruebas de software

Características que deben tener los ambientes

- Debe residir en un computador distinto al personal del desarrollador. Preferiblemente en un servidor compartido por los equipos de pruebas de software.

- Utilizar nombres de dominio (no direcciones IP) distintos a los de producción y desarrollo para evitar confusión del personal de pruebas.

- Ser lo más similarmente posible al ambiente de producción, incluyendo:

    - Aplicaciones locales, cliente servidor, web, etc.
    - Configuración de servidores.
    - Manejadores de Bases de datos de producción, de las mismas versiones.
    - Configuración de base de datos.
    - Cuentas de usuario de sistema operativo, bases de datos y aplicaciones con la misma configuración y privilegios de acceso.

- Si no es posible tener instalado el mismo manejador de base de datos que en producción y desarrollo, automatizar la propagación de cambios de un ambiente a otro (Existen herramientas de software que lo permiten).

- Instalar las herramientas de gestión de casos de prueba y gestión de incidencias en una maquina o servidor distinto al del ambiente de pruebas integrales, compartido por los equipos de pruebas.

- Capacidad de servir a múltiples audiencias, por ejemplo administradores de sistemas, usuarios finales, desarrolladores. En todos los casos sólo para ejecución de pruebas.

- Debe apoyarse en herramientas de control de versiones, especialmente cuando existen múltiples proyectos en paralelo probando distintas funcionalidades.

Restricciones a aplicar a los ambientes

No deben ser utilizados para actividades de desarrollo o producción.

- Los desarrolladores no deben poseer privilegios de acceso de modificación de ningún tipo en ambiente de pruebas, esto para evitar cambios en la configuración no informados.

- No posee herramientas de software o permisos de acceso especiales para ejecutar desarrollos de software, en su lugar, tiene una configuración similar o igual a la de producción.

- Sólo el administrador se encarga de instalar, actualizar o desplegar para el ambiente de pruebas, nunca el desarrollador directamente.

- Requiere de estrictos controles de cambio que mantengan rastro de las modificaciones en la configuración, para así asegurar resultados controlados y también que el ambiente sea similar a producción. Cualquier ciclo de pruebas que este en proceso puede quedar invalidado por cambios en la configuración (y por ende tener que repetirse).

- Una versión se instala en producción sólo después que ha sido instalada y probada en ambiente de pruebas integrales.

Investigación y entrenamiento

- Definir las especificaciones de ambientes de prueba que necesita el equipo.
- Antes de iniciar cualquier ciclo de pruebas, deben ejecutarse actividades para verificar que el ambiente ha sido configurado adecuadamente.
- Restricciones:
  - Sólo el administrador se encarga de instalar, actualizar o desplegar para el ambiente de pruebas, nunca el desarrollador directamente.
  - Una versión se instala en producción sólo después que ha sido instalada y probada en ambiente de pruebas integrales.

Testing de función: Enfocada en conseguir informes sobre la aptitud de las funciones, los métodos y los servicios que componen el ERP.

Testing de seguridad: Ideada para comprobar la seguridad de los datos que maneja el ERP para Pymes, es decir, los datos de la empresa.

Testing de volumen: Dirigida a medir la velocidad del ERP para procesar grandes cantidades de información en la base de datos, bien sea en entrada o en salida.

Diseño de casos de prueba.

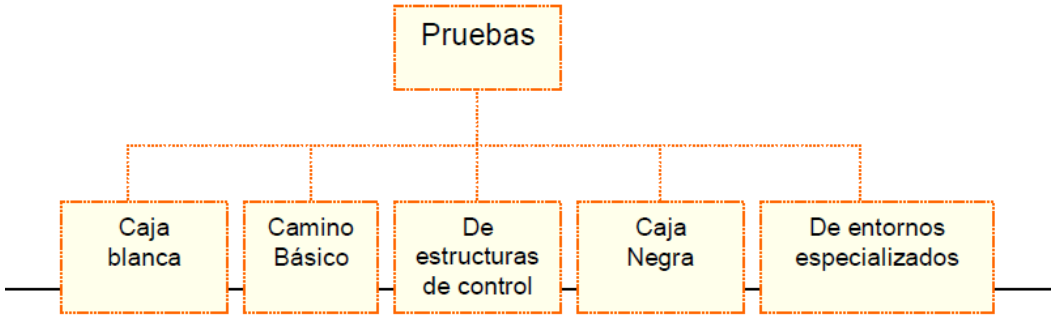
Cualquier producto software puede aprobarse de una las siguientes formas:

1. Conociendo la función para la que fue diseñado el producto.

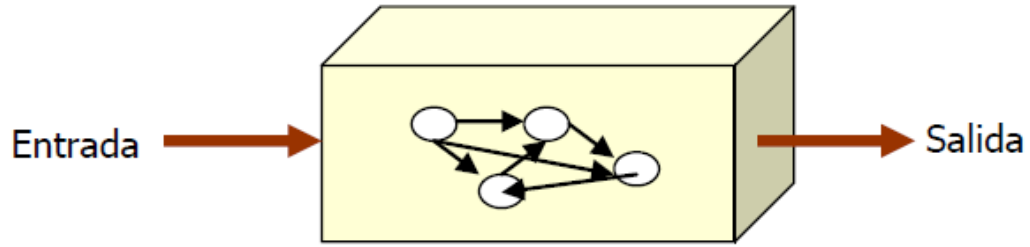
· Se pueden utilizar pruebas para: comprobar su función operativa y buscar errores de cada función.

2. Conociendo el funcionamiento del producto.

· Se pueden utilizar pruebas para: comprobar que las operaciones esta de acuerdo con las especificaciones y para comprobar que los componentes internos funcionan de forma adecuada.


Pruebas
Caja blanca
Camino Básico
De estructuras de control
Caja Negra
De entornos especializados

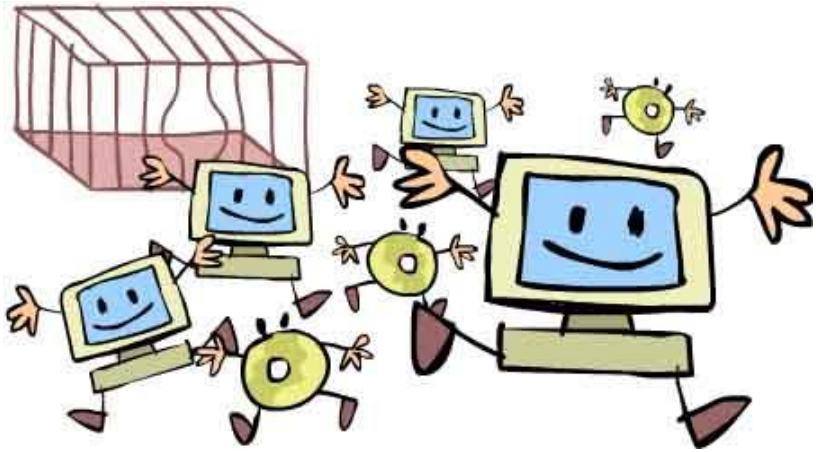
Estructura


Entrada
Salida

Administración de defectos

- En toda industria es importante siempre tener en claro que el principal problema a evitar son los defectos, nada molesta más a un cliente que no recibir lo esperado. En un proyecto de software aquello que se desea anticipar siempre es el buen uso del tiempo para entregar en buena forma el proyecto.

Características de los defectos



- Estos tienen diferentes características, pero los más importantes son
- 1.- crecen de forma exponencial ya que cada paso o proceso tiene un porcentaje de defecto y es una posibilidad que se multipliquen los problemas por cada paso que se da.
- 2.- Por errores humanos que terminan afectando al código ya que no se declaran variables de forma consistente.

Estrategias contra los defectos

- Revisar los productos durante cada uno de los pasos (pruebas unitarias)
- Hacer checklists
- Trabajar de la mejor manera posible
- Aprender de nuestros errores y mejorarlos

Validación de deployment.

- Tiene por objetivo asegurar que una aplicación se podrá implantar en un entorno

Entradas

•Software de la aplicación
•Scripts de base de datos

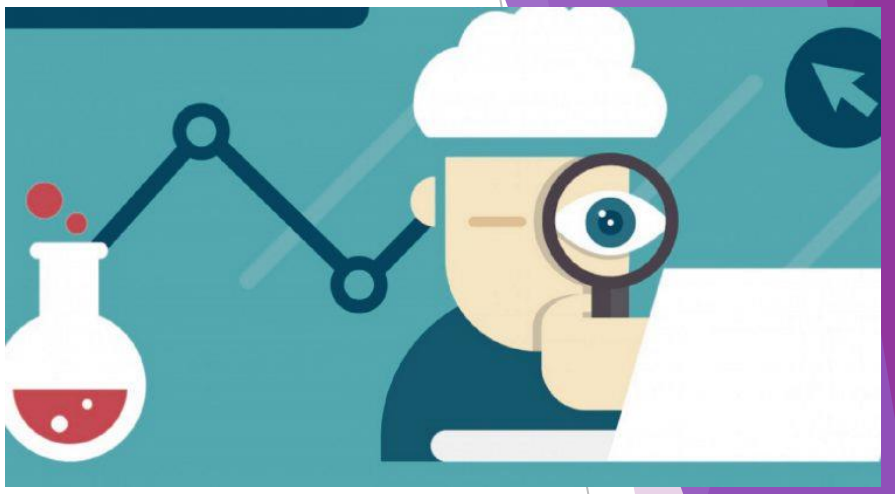
Salidas

•Documento Informe de Revisión de la documentación necesaria para la instalación y configuración
•Registro de los defectos detectados en el proceso, a través de la herramienta de Gestión de Defectos.

Actores implicados

•Equipo de Proyecto
•Equipo de Gestión de Servicios
•Equipo de Testing
•Equipo de Sistemas de Información.

El día a día de un tester.



Los probadores de software planifican y llevan a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente.

- Escribir y ejecutar scripts de prueba.
- Realizar pruebas manuales y automatizadas.
- Escribir informes de fallos.
- Revisar la documentación.
- Proporcionar garantía de calidad.
- Detectar potenciales fallos.
  Trabajar en múltiples proyectos a la vez.
- Supervisar aplicaciones y sistemas de software.

Realese

Pre-Alfa:
todas las actividades realizadas durante el proyecto de software antes de las pruebas formales.

Alfa:
Es la primera versión completa del programa, la cual es enviada a los verificadores para probarla.

Versión Candidata Definitiva:
Versión Final.

Beta:
Versión completa de un programa informático que es posible que sea inestable pero útil.

Sign Off