

ppt: Gestión y Control de Cambios

¿Que origina un cambio?

- Personalización
- Defectos
- Nuevas Funciones (mantenimiento)
- Cambios Externos (adaptación)

Administrar Solicitudes de Cambios:

Establecer estándares para un proceso consistente de control de cambios, e informar a los stakeholders del impacto y costo.

Registros:

Control de cambios:

- Código
- Documentación
- Base de datos

Auditoria de Configuración:

- Fecha
- Problemas detectados y recomendaciones

Actas de las reuniones de CCC (comité de creación de capacidades):

- Asistentes
- Propósito
- Disposiciones

Informes:

Tipos según previsión:

- Planificados
- Bajo demanda

Tipos según complejidad:

Directo: contenido de los registros

- Inventario de ECS
- Estado de los cambios

Indirectos

- Diferencias entre versiones

Políticas (Reglas):

- Establece el conjunto de Estados, Acciones y Reglas para el Proceso de Control de Cambios.

- Las Reglas establecen las acciones que cada Rol puede realizar para el cambio de estado de las peticiones.
- Establece para cada Acción los campos que deben ingresarse (obligatorios, opcionales o sólo lectura).

Estados y Acciones:

Estado

Estado de una petición de cambio en un momento dado y se modifica por las acciones del Usuario.

Acción

Actividad que cambia el Estado de una petición en un estado a otro.

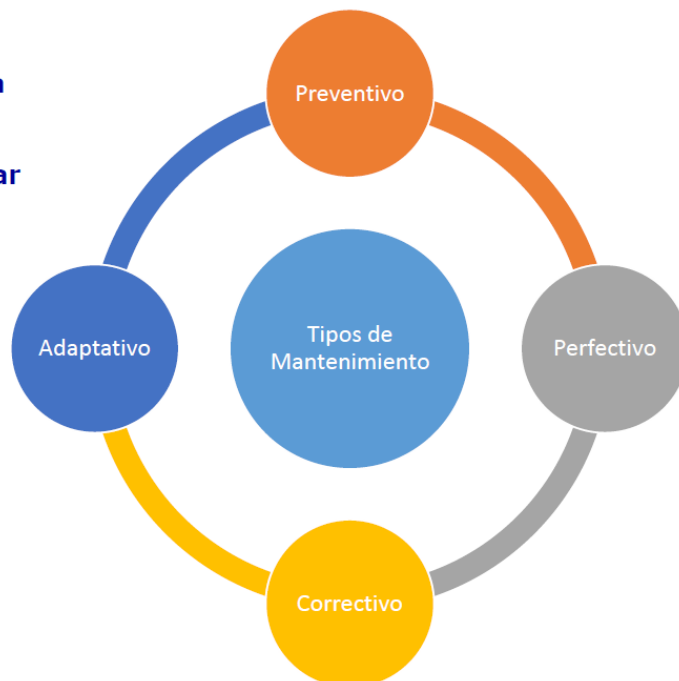
ppt: Mantenimiento de Software

Su objetivo principal es modificar el producto de software existente preservando si integridad, incluyendo la migración y retirada del producto software.

Tipos de Mantenimiento



la medida cualitativa de la facilidad para comprender, corregir y adaptar o mejorar el software.



La mantenibilidad es un atributo de calidad del software.

Correctivo:

Permite realizar la corrección de errores o fallas encontradas en el software. Su propósito consiste en modificar el software para corregir defectos. Es el más común, ya que es de carácter **urgente**.

Adaptativo:

Incluye modificaciones requeridas debidas al cambio de la configuración. Garantiza cambiar al software para acomodarlo a aquellos cambios que se producen en su entorno de operación. Puede ser por un cambio de versión, sistema operativo o componente (hardware).

Preventivo:

Incluye incorporaciones, modificaciones y eliminaciones necesarias en un producto para cubrir el cambio en las necesidades del usuario. Permite detectar y corregir fallas latentes en el producto software antes de que se manifiesten de manera efectiva.

Perfectivo:

Acciones para mejorar la calidad interna del software

- Reestructuración del código
- Definición más clara del sistema
- Optimización del rendimiento y eficiencia

Cambios a fin de que se pueda corregir, adaptar y mejorar de una manera mas fácil. Se busca mejorar el desempeño y la facilidad de mantenimiento del software.

Las 8 reglas de Lehman

1. Cambio continuo
2. Complejidad creciente
3. Autorregulación
4. Conservación de la estabilidad organizacional
5. Conservación de la familiaridad
6. Crecimiento continuo
7. Calidad decreciente
8. Sistema retroalimentativo

Sistema E-Type: son sistemas de software que solucionan un problema o implementan una aplicación para computadoras en el mundo real.

Ley 1- Cambio continuo

Un programa e-type debe continuamente adaptarse sino este se transforma en un software poco satisfactorio.

Ley 2 – Complejidad creciente

A medida que un programa evoluciona su complejidad aumenta a menos que se realice un trabajo de mantenimiento o se reduzca su complejidad.

Ley 3 – Autorregulación

Se balancea el desarrollo del software para lograr un crecimiento y estabilización controlada.

Ley 4 - Conservación de la estabilidad organizacional

Conduce a la estabilización del producto en su proceso de evolución.

Ley 5 – Conservación de la familiaridad

Involucramiento de todos los participantes en el desarrollo y evolución del producto.

Ley 6 – Crecimiento continuo

El contenido funcional del producto de software tiene que incrementarse para mantener la satisfacción de los usuarios en el tiempo.

Ley 7 - Calidad decreciente

Los programas E-Type se perciben como productos que decrementan su calidad a menos que sean rigurosamente mantenidos y adaptados a su entorno cambiante.

Ley 8 - Sistema retroalimentativo

Los procesos de programación de sistemas e-type son multi-lazos, con retroalimentación a diferentes niveles, de esta forma, pueden mejorar y modificarse satisfactoriamente. (FeedBack)

ppt: Auditoria de software

Auditoria: Emisión de una opinión profesional sobre si el objeto sometido a análisis presenta adecuadamente las condiciones que han sido prescritas.

Tipo de auditoria: Informática

- Objeto: Sistemas de aplicación, recursos informáticos, planes de contingencia, etc.
- Propósito: Operatividad eficiente y según normas establecidas

Consultoría: asesoramiento sobre qué y cómo se deben llevar adecuadamente una actividad para obtener fines deseados.

Clases de consultorías: Informática

- Contenido: asesoramiento
- Objetivo: aplicaciones, planes de contingencia
- Finalidad: desarrollo, diseño e implementación

Comparación entre Auditoria y Consultoría

Item	Auditoría	Consultoría
Contenido	Una opinión	Dar asesoramiento o consejo
Condición	Profesional	De carácter especializado
Justificación	Sustentada en ciertos procedimientos	En base a un examen o análisis
Objeto	Información obtenida en un cierto soporte	La actividad sometida a consideración
finalidad	Determinar si presenta adecuadamente la realidad o responde a las expectativas atribuidas (su finalidad)	Establecer la manera de llevarla a cabo adecuadamente

La auditoría informática:

Conjunto de procedimientos y técnicas para evaluar y controlar total o parcialmente un sistema informático. Con el fin de proteger el activo (información) o recursos, verificar si sus actividades se desarrollan eficientemente y cumplan con la normativa informática y general existente en la empresa y para conseguir la eficacia exigida en el marco de la organización.

ppt: Pruebas

Verificación: el proceso de evaluar un sistema o sus componentes para determinar si satisfacen las condiciones impuestas al comienzo de una fase.

Validación: el proceso de evaluación de un sistema o de uno de sus componentes al final del proceso de desarrollo para determinar si satisfacen los requisitos del usuario.

Relación entre defecto – falla – error

Error:

Acción humana que produce un resultado incorrecto (error de programación).

Defecto:

Desperfecto en un componente o sistema que puede causar un fallo en una función requerida. (sentencia o definición de datos incorrectos).

Fallo:

Manifestación física o funcional de un defecto (resultado no esperado).

Nota:

- Descubrir un error es el éxito de una prueba
- No se pueden probar todas las posibilidades de su funcionamiento

Enfoque de diseño de pruebas

Caja blanca: se centra en la estructura interna del programa (analiza los caminos).

Caja Negra: se centra en la entrada, funciones, salida.

Caja gris: crear casos de pruebas en base a modelos aleatorios o estadísticos.

Pruebas estructurales

Diseño de casos de pruebas en el que se valida la mayor cantidad de pasos para descubrir defectos en cada avance.

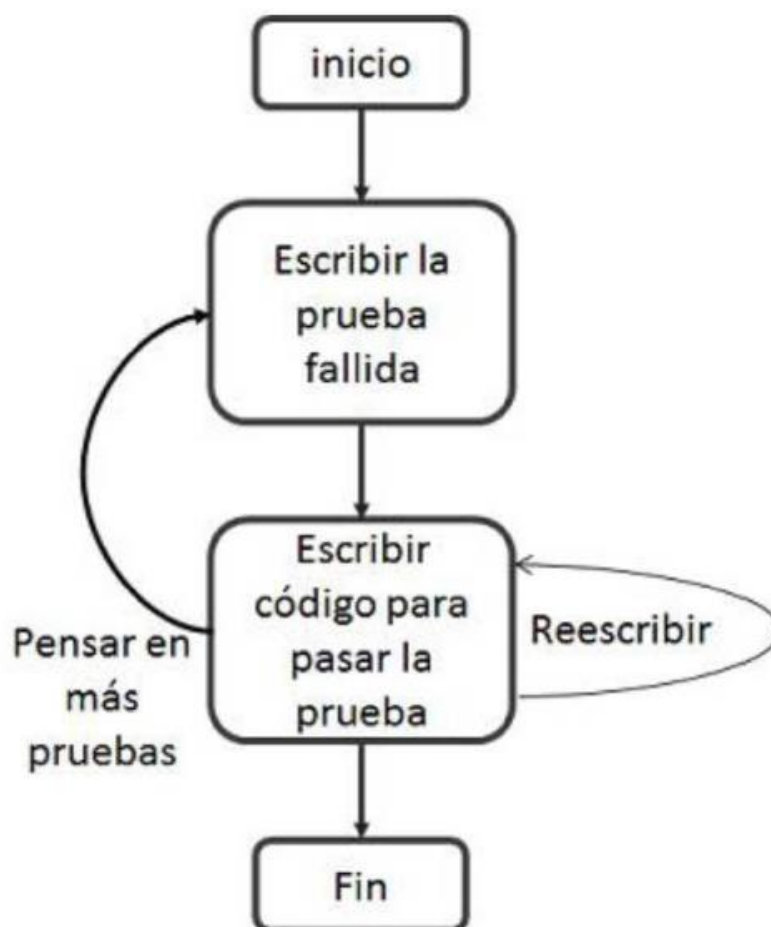
Pruebas funcionales

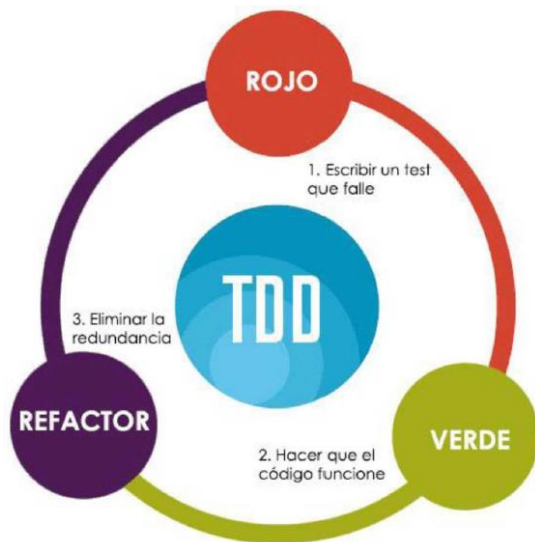
Se centra en las funciones, entradas y salidas. Para una determinada entrada se deben ejecutar funciones que retornen un resultado esperado.

Pruebas Aleatorias

Se utilizan programas generadores automáticos de casos de prueba. Crean datos de entrada que podrían parecer en la práctica y los testean de forma repetitiva.

Flujo de trabajo de Test-Driven Development (TDD)





Ventajas del TDD:

- La calidad del software aumenta
- Código altamente reutilizable
- El equipo de trabajo se une y fomenta la confianza.
- Multiplica la comunicación
- Incrementa la productividad
- Uno se marcha con la sensación de que el trabajo está bien hecho

TDD vs Pruebas al final:

- TDD requiere que se escriba la prueba primero y luego el código para pasarla por la prueba.
- Los programadores usan una varianza de hacer el código primero y luego someterlas a las pruebas.