INSTITUTO POLITECNICO NACIONAL

CENTRO DE ESTUDIOS CIENTIFICOS Y TECNOLOGICOS N. 9 "JUAN DE DE DIOS BATIZ"

INTRODUCCIÓN A LOS SISTEMAS DISTRIBUIDOS

REVISIONES E INSPECCIONES
&
MEDICIÓN Y MÉTRICAS DEL SOFTWARE

EQUIPO 10

Cedillo Lopez Erick Omar
Granados Martínez Pablo Daniel
Hernández Martínez Brian Arturo
Mejia Cruz Josue Dario
Tellez De La Cruz Esaul

REVISIONES E INSPECCIONES

¿Que Son?

Son actividades QA (ASEGURAMIENTO DE LA CALIDAD) que comprueban la calidad de los entregables. Puede implicar corregir bugs de software, refactorizar el software o reescribir documentos.

¿A Que Se Refiere?

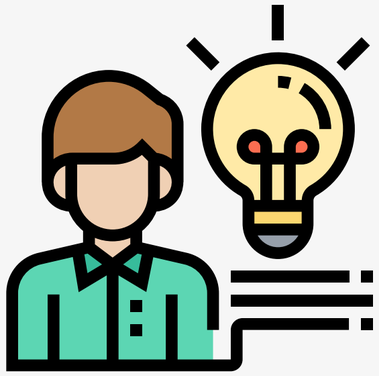
- Examinar El Software
- Examinar La Documentación
- Examinar Registros Del Proceso Para Descubrir Errores u Omisiones
- Observar Si Se Siguieron Los Estandares De Calidad

Se buscan problemas potenciales y falta de conformidad en los estandares.

Se realizan juicios sobre la calidad del entregable o proyecto

REVISIONES E INSPECCIONES

¿Para Que Sirven?



Los administradores del proyecto usan sus valoraciones respecto al entregable para tomar decisiones de planeación y asignar recursos al proceso de desarollo para asegurarse de que el proyecto a entregar sea util en tiempo y forma.
Es necesario encontrar los errores para garantizar una participación constructiva .

PROCESO

- Actividades Previas A La Revisión

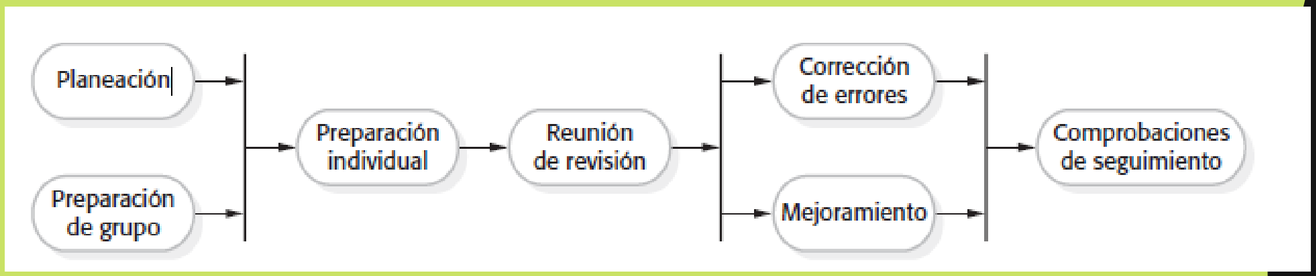
Establecer un equipo de rebición, organizar un tiempo , destinar un lugar para llevar a cabo la revisión , distribuir los documentos a revisar .

- Reuinión De Revisión

El autor de dicho objeto a revisión repasará el objeto junto al equipo destinado a revisar , debe ser una revisión corta. Otro compañero registra las decisiones o acciones ; se tiene que garantizar que los errores u omisiones comentados sean considerados, para concluir el equipo firma el registro de los comentarios y decisiones acordadas .

- Actividades Posteriores A La Revisión

Deberan ser tratados todos los problemas comentados en la revisión y el equipo de revision estara al pendiente que se contemplen tal cuales .



Clase de falla	Comprobación de inspección
Fallas de datos	• ¿Todas las variables del programa se inician antes de usar sus valores? • ¿Todas las constantes tienen nombre? • ¿La cota superior de los arreglos es igual al tamaño del arreglo o Valor – 1? • Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador? • ¿Existe alguna posibilidad de desbordamiento de buffer?
Fallas de control	• Para cada enunciado condicional, ¿la condición es correcta? • ¿Hay certeza de que termine cada ciclo? • ¿Los enunciados compuestos están correctamente colocados entre paréntesis? • En caso de enunciados, ¿se justifican todos los casos posibles? • Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?
Fallas de entrada/salida	• ¿Se usan todas las variables de entrada? • ¿A todas las variables de salida se les asigna un valor antes de que se produzcan? • ¿Entradas inesperadas pueden causar corrupción?
Fallas de interfaz	• ¿Todas las llamadas a función y método tienen el número correcto de parámetros? • ¿Los tipos de parámetro formal y real coinciden? • ¿Los parámetros están en el orden correcto? • Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de memoria compartida?
Fallas de gestión de almacenamiento	• Si se modifica una estructura vinculada, ¿todos los vínculos se reasignan correctamente? • Si se usa almacenamiento dinámico, ¿el espacio se asignó correctamente? • ¿El espacio se cancela explícitamente después de que ya no se requiere?
Fallas de gestión de excepción	• ¿Se tomaron en cuenta todas las posibles condiciones de error?

MEDICIÓN Y MÉTRICAS DEL SOFTWARE

¿Que Son?

Es la linea de referencia para valorar la efectividad de la herramienta. Se comparan componentes , sistemas o procesos .

¿A Que Se Refiere?

La medición es para realizar juicios de la calidad del software al usar medición de software, un sistema podría valorarse preferentemente mediante un rango de métricas y, a partir de dichas mediciones, se podría inferir un valor de calidad del sistema y si el software alcanzó un umbral de calidad requerido, entonces podría aprobarse sin revisión.

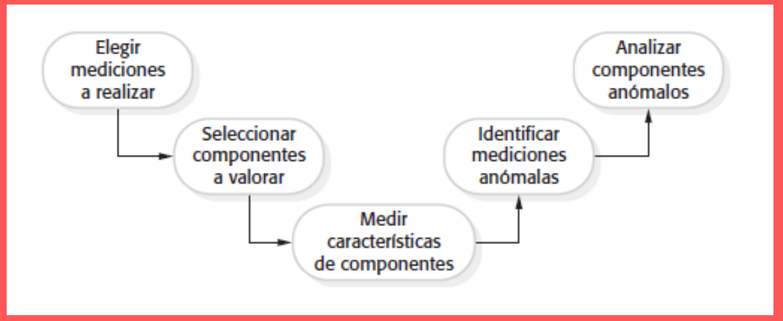
MEDICIÓN Y MÉTRICAS DEL SOFTWARE

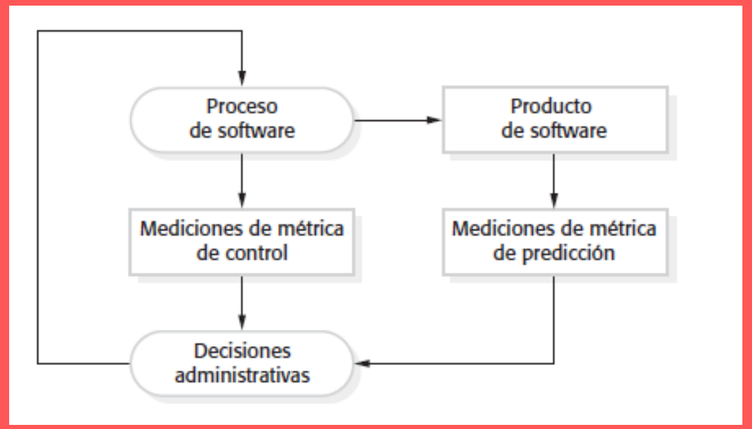
¿Para Que Sirven?

La medición del software se ocupa de derivar un valor numérico o perfil para un atributo de un componente, sistema o proceso de software.
Se comparan dichos valores unos con otros y en base a los estandares de la empresa se pueden sacar conclusiones sobre la calidad del software y valorar la efectividad de:

- Procesos
- Herramientas
- Metodos De Software

Proceso





1. Elegir las mediciones a realizar
Deben formularse las preguntas que la medición busca responder, y definir las mediciones requeridas para responder a tales preguntas.

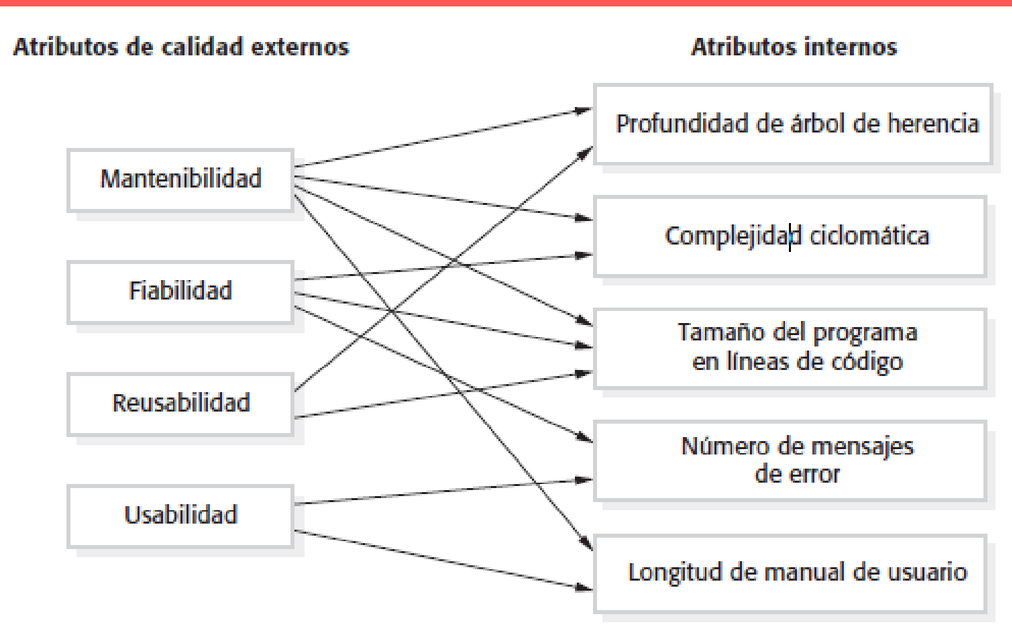
2. Seleccionar componentes a valorar
Probablemente usted no necesite estimar valores métricos para todos los componentes en un sistema de software, ya que en ocasiones podrá seleccionar una muestra representativa de componentes para medición, que le permitirá realizar una valoración global de la calidad del sistema.

3. Medir las características de los componentes
Se miden los componentes seleccionados y se calculan los valores de métrica asociados.

4. Identificar mediciones anómalas
Después de hacer las mediciones de componentes, se comparan entonces unas con otras y con mediciones anteriores que se hayan registrado en una base de datos de mediciones.

5. Analizar componentes anómalos
Cuando identifique los componentes con valores anómalos para sus métricas seleccionadas, debe examinarlos para decidir si dichos valores de métrica anómalos significan que la calidad del componente se encuentra o no comprometida.

Métrica orientada a objetos	Descripción
Métodos ponderados por clase (*weighted methods per class*, WMC)	Éste es el número de métodos en cada clase, ponderado por la complejidad de cada método. Por lo tanto, un método simple puede tener una complejidad de 1, y un método grande y complejo tendrá un valor mucho mayor. Cuanto más grande sea el valor para esta métrica, más compleja será la clase de objeto. Es más probable que los objetos complejos sean más difíciles de entender. Tal vez no sean lógicamente cohesivos, por lo que no pueden reutilizarse de manera efectiva como superclases en un árbol de herencia.
Profundidad de árbol de herencia (*depth of inheritance tree*, DIT)	Esto representa el número de niveles discretos en el árbol de herencia en que las subclases heredan atributos y operaciones (métodos) de las superclases. Cuanto más profundo sea el árbol de herencia, más complejo será el diseño. Es posible que tengan que comprenderse muchas clases de objetos para entender las clases de objetos en las hojas del árbol.
Número de hijos (*number of children*, NOC)	Ésta es una medida del número de subclases inmediatas en una clase. Mide la amplitud de una jerarquía de clase, mientras que DIT mide su profundidad. Un valor alto de NOC puede indicar mayor reutilización. Podría significar que debe realizarse más esfuerzo para validar las clases base, debido al número de subclases que dependen de ellas.
Acoplamiento entre clases de objetos (*coupling between object classes*, CBO)	Las clases están acopladas cuando los métodos en una clase usan los métodos o variables de instancia definidos en una clase diferente. CBO es una medida de cuánto acoplamiento existe. Un valor alto para CBO significa que las clases son estrechamente dependientes y, por lo tanto, es más probable que el hecho de cambiar una clase afecte a otras clases en el programa.
Respuesta por clase (*response for a class*, RFC)	RFC es una medida del número de métodos que potencialmente podrían ejecutarse en respuesta a un mensaje recibido por un objeto de dicha clase. Nuevamente, RFC se relaciona con la complejidad. Cuanto más alto sea el valor para RFC, más compleja será una clase y, por ende, es más probable que incluya errores.
Falta de cohesión en métodos (*lack of cohesion in methods*, LCOM)	LCOM se calcula al considerar pares de métodos en una clase. LCOM es la diferencia entre el número de pares de método sin compartir atributos y el número de pares de método con atributos compartidos. El valor de esta métrica se debate ampliamente y existe en muchas variaciones. No es claro si realmente agrega alguna información útil además de la proporcionada por otras métricas.



¡¡¡GRACIAS POR VER!!!