



aprenderaprogramar.com

Calidad del software. Métricas, CMMI y fiabilidad de aplicaciones (1ª parte) (DV00103A)

Sección: Divulgación

Categoría: Tendencias en programación

Fecha revisión: 2029

Autor: César Krall

Resumen: Este artículo explica cuestiones básicas sobre la calidad del software y sobre qué son las métricas, norma CMMI, etc. Resume y comenta una conferencia impartida por Ramiro Carballo (Ingeniero de la empresa GESEIN) en la Escuela de Ingeniería Informática de la Universidad de Sevilla.

CALIDAD DEL SOFTWARE

El software es un producto como cualquier otro, y por tanto podemos hablar de software de buena calidad y software de mala calidad. La calidad del software comprende distintos aspectos como estética (que sea agradable a la vista), funcionalidad (que sea fácil de usar), eficiencia (que ejecute con rapidez y precisión los procesos), etc.



Lo que distingue al software de otros productos industriales es que no es de naturaleza material, no se puede tocar. Por tanto no resulta viable hacer una valoración del mismo en base a una impresión rápida o análisis del aspecto ni en base al coste de materiales componentes.

MÉTRICA

Históricamente se habló de métrica en referencia a los sistemas que existían para escribir versos diferenciados en base al número de sílabas que contenía cada verso, así como en referencia al estudio y “medición” de la cantidad de sílabas y estrofas que contenían los versos. En informática, el término métrica hace referencia a la medición del software en base a parámetros predeterminados, como puede ser el número de líneas de código de que consta o el volumen de documentación asociada. A veces en vez de hablar de métrica se usa el término “Indicadores” del software. Algunos ingenieros lo usan como sinónimos mientras que otros les atribuyen significados distintos.

Algunas métricas o indicadores pueden ser:

- a) Índice de productividad \cong tamaño / esfuerzo \cong líneas de código generado / horas trabajadas.
- b) Tasa de defectos \cong defectos / tamaño \cong número de errores / líneas de código generadas.

¿QUÉ ES LA ISO 9001 EN RELACIÓN AL SOFTWARE?

La ISO 9001 es una normativa de cumplimiento opcional para el aseguramiento de la calidad en las empresas. Esta norma trata de definir los procesos que se siguen dentro de la empresa para estandarizarlos y controlarlos. Es de carácter genérico. El cumplimiento de la norma ISO 9001 no garantiza que se esté controlando que la calidad del producto final sea buena. Simplemente garantiza que la empresa ha adoptado una organización definida y controlada.

¿QUÉ SON LOS MODELOS CMMI?

CMMI son las siglas de un modelo utilizado en el ámbito de la informática para evaluar si una empresa mantiene ciertos niveles de calidad en relación al software. CMMI es un estándar que cuenta con gran número de empresas seguidoras, pero hay otras normas y modelos que son seguidos por otras empresas, como CMM, ISO 9001, SPICE, PSP, TSP, ISO 20.000, ITIL, etc.

Una empresa que quiera acreditarse como cumplidora del modelo CMMI habrá de pasar una evaluación. Existen distintos niveles dentro del modelo CMMI. Según el nivel en que se encuentre una empresa, tendrá que cumplir con requisitos más o menos exigentes.

Para implantar CMMI no hace falta ser una gran empresa. Puede ser implantado desde en una pequeña empresa con cinco empleados hasta en una multinacional con 50.000.

El modelo CMMI es gratuito. Está en internet y se puede descargar en formato pdf o doc. En la web de la Asociación Española de la Calidad (aec.es) podemos descargar una guía de CMMI para pequeñas organizaciones; en esta web, que abarca calidad en diversos sectores además del informático, se publican noticias interesantes sobre calidad en el software y las tecnologías de la información.

Hemos dicho que las métricas servían en informática para hacer mediciones del software. Cuando se implanta un sistema de calidad como CMMI, se usan las métricas para comprobar que se producen cambios reales en el software que produce la empresa. Si implantar un sistema de este tipo no derivara en mejoras reales que puedan medirse, muchas empresas optarían por no utilizarlos.

Muchas empresas evalúan si una nueva tecnología les aporta algo utilizando las métricas. Por ejemplo, si un proyecto requería 750.000 líneas de código y con una nueva tecnología pasa a requerir 500.000 posiblemente se haya mejorado en eficiencia.

PRUEBAS EN EL SOFTWARE: DE LA TEORÍA A LA PRÁCTICA

Cuando el desarrollo de software adquiere forma comienza el desarrollo de pruebas. La inversión en pruebas es importante. En la fase de pruebas se utilizan los denominados casos de prueba: una secuencia de pasos que debe llevarnos a un resultado esperado. Cuando el resultado se desvía de lo esperado decimos que el resultado ha sido "un fallo".

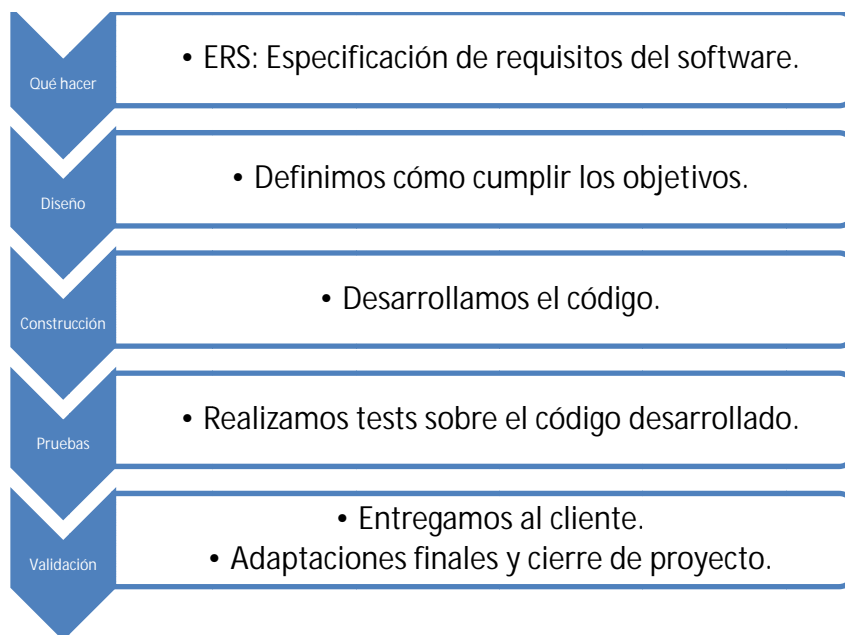
El tiempo que se consume en pruebas de software se traduce en costes económicos para la empresa, ya que supone tanto gastos de personal encargado de las pruebas como demora en la entrega del producto al cliente. Para optimizar estos procesos debe existir una metodología de pruebas y un registro de los resultados de prueba.

En teoría, nunca debe entregarse un proyecto que no haya sido sometido a pruebas rigurosas. En la práctica, normalmente los programadores trabajan con presión y el software se entrega al cliente antes de que se haya realizado un proceso de pruebas adecuado.

Desde el punto de vista teórico, lo ideal es que exista un equipo de prueba independiente del equipo de desarrollo y que genere una incidencia por cada defecto que se encuentre en el software. En la práctica, es frecuente que los mismos desarrolladores realicen las pruebas.

La persona que se especializa en la realización de pruebas se suele llamar "tester" y los errores que encuentra "bugs", todo ello derivado de la terminología anglosajona. Si en un proyecto tenemos un tester, si decidimos poner 2 tester no significa que se vayan a encontrar el doble de errores porque hay un cierto grado de paralelización o solape. Hay que decidir cuántos tester y cuánto tiempo de trabajo de testers hay que dedicarle a cada proyecto.

LAS PRUEBAS Y LAS MÉTRICAS EN EL CICLO DE VIDA DEL SOFTWARE



Cuando el cliente nos da una especificación de requisitos del software (ERS) se procede a cuantificar el tamaño y complejidad de lo que nos piden para poder hacer un presupuesto. La técnica más utilizada para estimar el tamaño es la técnica del punto función, una técnica que trata de enumerar las consultas, datos, informes, etc. que van a ser necesarios para obtener el producto terminado.

Las métricas nos permiten saber, entre otras cosas, el número o importancia de los errores que se detectan en los tests o correspondientes a reclamaciones recibidas del cliente. Si en cada proyecto medimos el grado de error con el tiempo tendremos un histórico que nos irá diciendo si vamos mejorando o no. También nos servirá para realizar predicciones sobre cómo el volumen de errores y tiempo de corrección que será necesario en nuevos proyectos antes de la fase de pruebas del mismo. En resumen, la información recopilada de cada proyecto nos servirá para el futuro.

Continuación del artículo en: DV00104A (busca este código en el buscador de aprenderaprogramar.com)

REFERENCIAS Y MÁS INFORMACIÓN

Este artículo resume y comenta la conferencia pública impartida por Ramiro Carballo, Ingeniero Informático de la empresa GESEIN, en el marco de las "Jornadas Imaginática: La informática del futuro", que tuvieron lugar en la Escuela Técnica Superior de Informática de la Universidad de Sevilla (España) y a las que tuvimos la oportunidad de asistir.

Gesein es una empresa con sede en Madrid que cuenta con unos 250 empleados. Cuentan con una sección de consultoría para la optimización del desarrollo del software. Trabajan en la implantación de modelos de procesos como ISO 9001, CMMI, etc.

Para más información: Asociación Española para la calidad: www.aec.es; www.gesein.com.