C a l i d a d d e s o f t w a r e ll

**1. Factores de calidad en SW**

Brenda Juárez Santiago

PROFESORA

Quinto Cuatrimestre

Índice

Contenido

[DIAGRAMA DE ASIGNATURA 3](#_Toc502921220)

[CAPITULO 1. Introducción a la calidad en el desarrollo de software 4](#_Toc502921221)

[1.1. Generalidades de la Calidad 4](#_Toc502921222)

[1.1.2. Norma 5](#_Toc502921223)

[1.1.3. Estándares. 5](#_Toc502921224)

[1.1.4. Procesos 5](#_Toc502921225)

[1.1.5. Modelos 6](#_Toc502921226)

[1.1.6. Institutos que regulan la calidad. 7](#_Toc502921227)

[1.2 Conceptos en el desarrollo de software 9](#_Toc502921228)

[1.3 Factores de Calidad según McCall 9](#_Toc502921229)

[1.4 Lista de factores: 9](#_Toc502921230)

[1.5 Factores de Calidad según Boehm 11](#_Toc502921231)

[1.6 Factores de Calidad según ISO 9126 12](#_Toc502921232)

[Referencias 14](#_Toc502921233)

# DIAGRAMA DE ASIGNATURA

CALIDAD EN EL DESARROLLO DE SOFTWARE



Generalidades de la calidad



Concepto de calidad en el

desarrollo de software



4

)

Técnicas de estimación



SUBTEMAS



MODULOS



1)

Introducción a la calidad

en el desarrollo de software



2)

Métricas de software



3)

Proceso personal de

desarrollo de software

(

).

PSP



Métricas de software



Tipos de métricas de calidad de

software

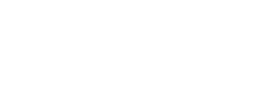
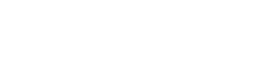
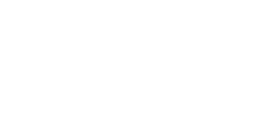


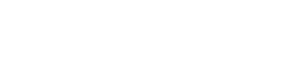
Diagrama de clases



Puntos de función



Puntos de caso de uso



Proceso personal de

desarrollo de software



Plantillas PSP

5

)

Modelos para el

aseguramiento de la

calidad de software

MOPROSOFT

CMMI

## CAPITULO 1. Introducción a la calidad en el desarrollo de software

## 1.1. Generalidades de la Calidad

Es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

Existen 3 puntos importantes de la definición de calidad de software:

1. Los requerimientos del software son los fundamentos desde los que se mide la calidad.

1. Los estándares específicos definen un conjunto de criterios de desarrollo que guían la forma de aplicación de la ingeniería de software.

1. Existen requerimientos implícitos que no se mencionan.

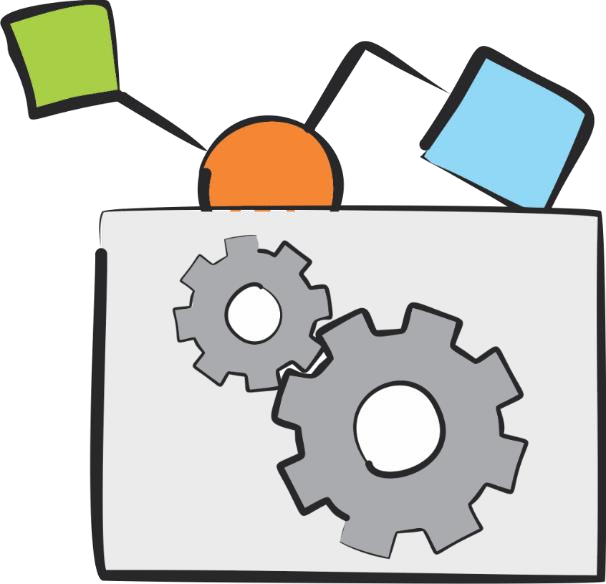
Un producto de alta calidad requiere menos mantenimiento y facilita tanto el desarrollo como el mantenimiento de la productividad. Con la medición de la calidad se pueden lograr estos objetivos. En lo que se refiere al mantenimiento, la medición de la calidad del software ayuda a identificar problemas de confiabilidad y a mejorar las técnicas para identificar las necesidades de mantenimiento. (Belgrano, s.f.)

## 1.2. Norma

A lo largo de los últimos años se han elaborado trabajos de investigación, normas y estándares, con el objetivo de crear modelos, procesos y herramientas de evaluación de la calidad del propio producto software. Precisamente para dar respuesta a estas necesidades nace la nueva familia de normas ISO/IEC 25000 conocida como SQuaRE (Software Product Quality Requirements and Evaluation), que tiene por objetivo la creación de un marco de trabajo para evaluar la calidad del producto software, sustituyendo a las anteriores ISO/IEC 9126 e ISO/IEC 14598 y convirtiéndose así en el referente a seguir.(Parra, 2014)**.**

## 1.3. Estándares.

Es un conjunto de reglas que deben cumplir los productos, procedimientos o investigaciones que afirmen ser compatibles con el mismo producto. Los estándares ofrecen muchos beneficios, reduciendo las diferencias entre los productos y generando un ambiente de estabilidad, madurez y calidad en beneficio de consumidores. (Ingeniería, s.f.)



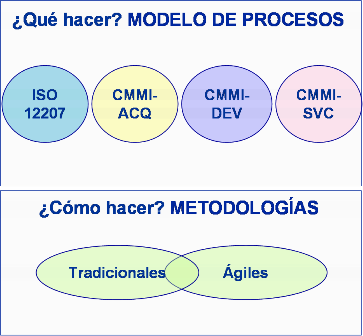
## 1.4. Procesos

Conjunto de prácticas secuenciales que son funcionalmente coherentes y reutilizables en la organización, implementación y administración de la ingeniería del software.

## 1.5. Modelos

Actualmente se han desarrollado modelos para la gestión de los procesos de software a nivel mundial con el fin de mejorar la productividad y la calidad. Uno de los modelos más difundidos es Modelo de Capacidad y Madurez Integrado (Capability Maturity Model Integration - CMMI). Este proporciona una orientación sobre el modo de realizar gestión a los procesos se clasifican en cuatro categorías ingeniería, gestión de proyectos, gestión de procesos y soporte, a la vez que se logra una evolución hacia la cultura de la ingeniería y una mejora continua.

Los modelos de calidad como CMMI for Development v1.2 [4] o ISO/IEC 15504:2003 se implementan en empresas con departamentos de desarrollo a partir de 1520 personas y requieren una gran cantidad de herramientas, de las cuales muchas de ellas tienen un costo elevado y el retorno de inversión se obtiene a medio-largo plazo; por lo tanto, para las MiPymes de software de nuestro país no es viable adoptar estos modelos. (Pereira, 2009)



## 1.6. Institutos que regulan la calidad.

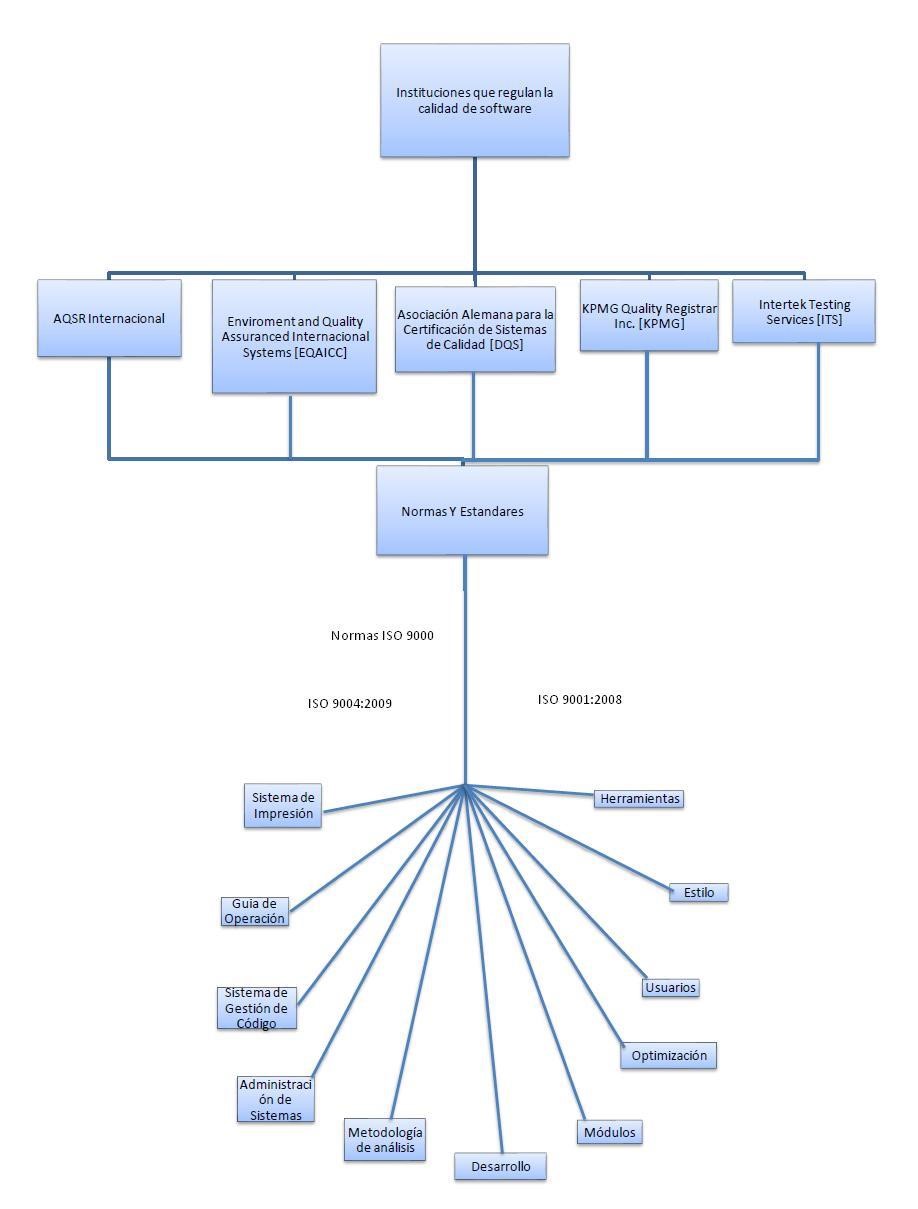
**ISO** Organización Internacional para la Estandarización.

**NORMEX** Sociedad Mexicana de Normalización y Certificación.

**IQC** International Quality Certification.

**GLC** Germanischer Lloyd Certificaction México.

(best007, 2011)



## 2.1 Conceptos en el desarrollo de software

## 2.2 Factores de Calidad según McCall

Loa factores desarrollados según el modelo de McCall, se centra en tres aspectos importantes de un producto de software:

* Sus características operativas.
* Su capacidad para soportar los cambios.
* Su adaptabilidad a nuevos entornos.

## 2.3 Lista de factores:

* **Corrección:**

**¿Hace lo que se le pide?**

Mide el grado en que un programa satisface sus especificaciones y consigue los objetivos del usuario.

* **Fiabilidad:**

**¿Lo hace de forma fiable todo el tiempo?**

Mide el grado en que se puede esperar que un programa lleve a cabo sus funciones esperada con la precisión requerida.

* **Eficiencia:**

**¿Qué recursos hardware y software necesito?**

Mide la cantidad de recursos de computadora y de código requerido por un programa para que lleve a cabo las funciones especificadas.

* **Integridad:**

**¿Puedo controlar su uso?**

Es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.

* **Facilidad de Uso:**

**¿Es fácil y cómodo de manejar?**

Es el esfuerzo requerido para aprender un programa e interpretar la información de entrada y de salida.

* **Facilidad de Mantenimiento:**

**¿Puedo localizar los fallos?**

Es el esfuerzo requerido para localizar y arreglar programas.

* **Facilidad de Prueba:**

**¿Puedo probar todas las opciones?**

Es el esfuerzo requerido para probar un programa.

* **Flexibilidad:**

**¿Puedo añadir nuevas opciones?**

Es el esfuerzo requerido para modificar un sistema operativo.

* **Portabilidad:**

**¿Podré usarlo en otra máquina?**

Es el esfuerzo requerido para transferir un software de un hardware o un entorno de sistemas a otro.

* **Reusabilidad:**

**¿Podré utilizar alguna parte del software en otra aplicación?**

Es el grado en que un programa (o partes de un programa) se puede reutilizar en otro.

* **Facilidad de Interoperación:** es el esfuerzo requerido para asociar un programa a otro.

## 2.4 Factores de Calidad según Boehm

El modelo que presenta Boehm presenta una jerarquía de características donde cada una de ellas contribuye a la calidad global. Se centra en:

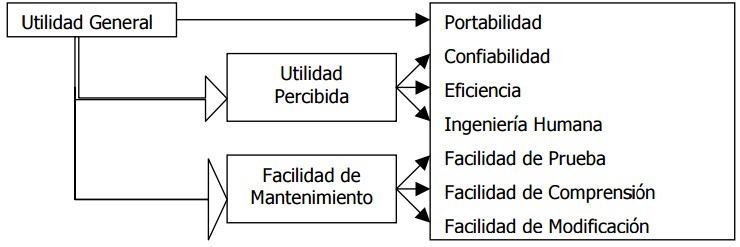
* Sus características operativas.

Factores según McCall

* Su capacidad para soportar los cambios.
* Su adaptabilidad a nuevos entornos.
* La evaluación del desempeño del hardware.

El modelo comienza con la utilidad general del software, afirmando que el software es útil, evitando pérdida de tiempo y dinero.

La utilidad puede considerarse en correspondencia a los tipos de usuarios que quedan involucrados. El primer tipo de usuarios queda satisfecho si el sistema hace lo que el pretende que haga; el segundo tipo es aquel que utiliza el sistema luego de una actualización y el tercero, es el programador que mantiene el sistema.

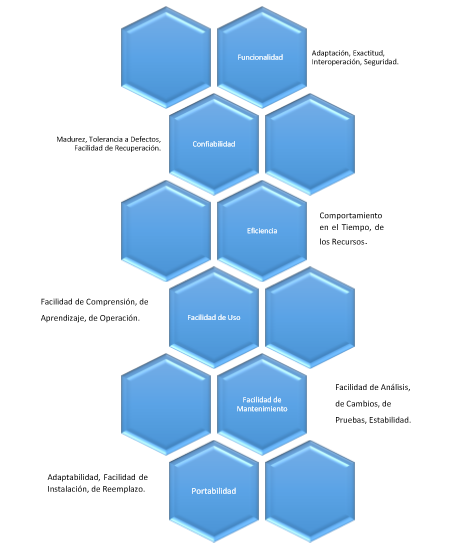


## 2.5 Factores de Calidad según ISO 9126

Es un modelo jerárquico con seis atributos especiales.

La diferencia con McCall y Boehm es que la jerarquía es estricta, es decir, que cada característica de la derecha solo está relacionada con un solo atributo del modelo. Las características de la derecha se relacionan con la visión del usuario.

* Funcionalidad............................... Adaptación, Exactitud, Interoperación, seguridad.
* Confiabilidad................................ Madurez, Tolerancia a Defectos, Facilidad de Recuperación.
* Eficiencia...................................... Comportamiento en el Tiempo, de los recursos.
* Facilidad de Uso........................... Facilidad de Comprensión, de Aprendizaje, de Operación.
* Facilidad de Mantenimiento......... Facilidad de Análisis, de Cambios, de Pruebas, Estabilidad.
* Portabilidad.................................. Adaptabilidad, Facilidad de Instalación, de Reemplazo**.**



# Referencias

Belgrano, U. (s.f.). *Universidad Belgrano* . Obtenido de http://www.ub.edu.ar/catedras/ingenieria/ing\_software/ubftecwwwdfd/calida dsw/calidad.htm

best007. (21 de 01 de 2011). *Blogdiario*. Obtenido de http://best007.blogspot.es/1295580434/calidad-en-el-desarrollo-delsoftware/

Ingeniería, F. d. (s.f.). *UNAM*. Obtenido de http://redyseguridad.fi-

p.unam.mx/proyectos/biometria/estandares/estandar.html

LLC, T. (2015). *CalidaddeSoftwareVeronica*. Obtenido de https://calidaddesoftwareveronica.wikispaces.com/Factores+que+determina n+la+calidad+del+software

Morón, U. d. (s.f.). CALIDAD DE SOFTWARE. Morón.

Parra, J. G. (2014). *Jornada sobre Calidad del Producto Software e ISO.* Santiago de Compostela: Xunta de Galacia.

Pereira, U. T. (08 de 2009).