

2019 .



CIINATIC
CONGRESO INTERNACIONAL
EN INNOVACIÓN Y APROPIACIÓN
DE LAS TECNOLOGÍAS DE LA INFORMACIÓN
Y LAS COMUNICACIONES

EXPLORING QUALITY MODELS FOR DESKTOP APPLICATIONS

EXPLORACIÓN DE MODELOS DE CALIDAD PARA APLICACIONES DE ESCRITORIO

Mg. Katerine Beleño Caselles, Ronald Palencia, Martha Doria Hernández

Universidad Popular del Cesar Seccional Aguachica, Departamento Sistemas e Informática, Programa de Ingeniería de Sistemas, Grupo de Investigación GIDEATIC.

Carrera 40 vía al Mar, Aguachica, Cesar, Colombia.

Telefax (5) 5654900-5657700-5655345. Cel 3175341554

E-mail: {katerinebeleno@unicesar.edu.co, ronaldpalenciaz@hotmail.com, marthadoriah27@gmail.com }.

Abstract— The quality of software is a relatively new issue that has emerged in Software Engineering, around which there are still many doubts in the application of the standards and models that many researchers have proposed. The importance of the subject lies in providing tools to know the state of the software that is developed. Today there are many software development companies that devote part of their resources to the subject.

Some authors have proposed that the solution to the problem of evaluating the software, goes through the decomposition of generic quality concepts easier to measure and evaluate, giving the name of Quality Model to this type of decomposition. (Cataldi, 2000).

This article intends to carry out an exploration of the different quality models and investigations that have been carried out to evaluate the quality of the software cataloged as desktop applications, highlighting the date in which they appear and the objective of the measurement. The models that will be analyzed will be focused on the internal quality of the software ie the product.

Keywords— software quality, product quality, software measurement, product quality, desktop applications.

Resumen— La calidad del software es un tema relativamente nuevo que ha emergido en la Ingeniería del Software, alrededor del cual aún hay muchas dudas en la aplicación de las normas y modelos que muchos investigadores han propuesto. La importancia del tema radica en proveer de herramientas para conocer el estado del software que se desarrolla. Hoy día son muchas las empresas desarrolladoras de software que dedican parte de sus recursos en el tema.

Algunos autores han propuesto que la solución al problema de evaluar el software, pasa por la descomposición de conceptos genéricos de calidad más sencillos de medir y evaluar, dándole el nombre de Modelo de Calidad a este tipo de descomposición. [1].

Este artículo pretende realizar una exploración de los diferentes modelos de calidad e investigaciones que se han realizado para evaluar la calidad del software catalogado como aplicaciones de escritorio, destacando la fecha en que aparecen y el objetivo de la medición. Los modelos que se analizarán serán enfocados a la calidad interna del software es decir al producto.

Palabras Claves— calidad de software, calidad del producto, medición de software, calidad del producto, aplicaciones de escritorio

INTRODUCCIÓN

El significado del término **modelo** (del italiano modello) reseña diferentes significados según el diccionario de la Real Academia Española (RAE). Entre ellos podemos destacar que un modelo es un arquetipo o punto de referencia para imitarlo o reproducirlo, o un ejemplar que se debe seguir e imitar por su perfección, o un esquema teórico, generalmente en forma matemática, de un sistema o una realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento.

En el presente documento se pretende presentar el estado del arte de los posibles modelos que han sido utilizados para la medición del producto software una vez éstos han sido desarrollados; aunque existen modelos de calidad que miden el proyecto, el proceso y el producto [2] en el presente documento se mencionarán los que tratan del producto software para aplicaciones de escritorio. En el caso de las mediciones para el producto existen modelos de calidad que se han venido desarrollando a través del tiempo y de los cuales se pueden mencionar entre los primeros el de McCall, FURPS, Gilb, Boehm, hasta llegar a la SQUARE 25000 [3]; los modelos y estándares

a nivel del producto surgen o se actualizan de acuerdo con la evolución tecnológica ocurrida [3]. La mayor parte de éstos modelos están basados en la norma 9126 [4].

Los modelos y estándares que se han estudiado hasta ahora se organizan en factores ó criterios, atributos, métricas o indicadores y explican cada uno de sus elementos a tener en cuenta en la evaluación de calidad del producto, éstos modelos no se especifican para fases del producto como análisis, diseño, implementación o mantenimiento; por ello es deber del grupo o investigador de calidad realizar la respectiva aplicación del modelo según el caso y la necesidad a todo el producto en general.

Los modelos estudiados mantienen la siguiente estructura a nivel general:

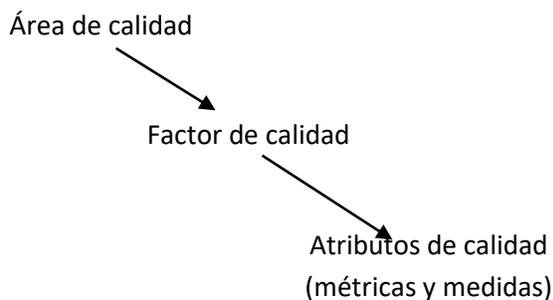


Figura 1. Jerarquía de la Metodología SAQE. Tomado de Scalone Fernanda.

En esta investigación se trató de dar una mirada a los modelos de calidad para el producto de aplicaciones de escritorio, con el objeto de clarificar cómo en la medida que ha venido evolucionando la tecnología han aparecido o han evolucionado algunos modelos existentes, algunos de ellos han sido propuestos con base en la 9126.

De otro lado, la información recolectada a partir de estos modelos sirvió como producto intermedio para el proyecto de investigación estado de la calidad interna de las aplicaciones de escritorio desarrolladas como proyecto de grado de la Universidad Popular del Cesar Seccional Aguachica aplicando la normas ISO/IEC 25000.

1. CALIDAD DEL SOFTWARE

[5] Pressman en su libro Ingeniería del software, define la calidad del software como “concordancia del software producido con los requisitos específicamente establecidos, con los estándares de desarrollo expresamente fijados y con los requisitos implícitos, no establecidos formalmente, que desea el usuario”. Según la IEEE, [6] Std. 610-1990, Diccionario de estándar de términos de computación, “es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Sea cual sea el tipo de aplicación que se desarrolle ya sea tradicional o aplicación web es deseable para el equipo desarrollador entregar un buen producto, para ello se encuentran en la literatura un número considerable de modelos de calidad.

2. MODELOS Y ESTÁNDARES DE CALIDAD DEL PRODUCTO SOFTWARE: APLICACIONES DE ESCRITORIO

Las aplicaciones de escritorio podrían pensar algunos que en la actualidad no están siendo muy utilizadas por todas las ventajas que ofrecen los entornos web, debido a que con éstas últimas no hay que preocuparse por su instalación, consumen menos recursos de la máquina, sólo requieren un navegador para su uso entre otras ventajas; sin embargo, hay situaciones donde una vez analizadas las necesidades de las empresas éstas son útiles y necesarias. Es por ello que se propone el estudio de los diferentes modelos que pueden servir para medir el estado de calidad de las mismas pues en la literatura no existe un modelo de medición del producto software específicamente para las aplicaciones de escritorio como si existen para las aplicaciones web.

Para medir el software y así conocer el estado de la calidad del mismo tanto las diferentes empresas de desarrollo como investigadores de diferentes latitudes han propuesto diferentes modelos o estándares para

brindar un marco que pueda ser una guía para las empresas y los desarrolladores de este producto como lo es el software. Entre ellos podemos mencionar:

Modelo de McCALL (1977)

Este es uno de los modelos clásicos para evaluar la calidad del software, fue desarrollado por MacCall [7] y de él se derivan otros como el de Boehm [3]; éste modelo está compuesto por 11 características o factores de calidad las cuales se organizan en tres categorías.

Categorías:

- Operación del producto.
- Revisión del producto.
- Transición del producto.

Características de Operación del producto: Corrección, Fiabilidad, Eficiencia, Integridad, Facilidad de uso.

Características Revisión del producto (Capacidad para soportar cambios): Facilidad, de mantenimiento, Flexibilidad, Facilidad de prueba.

Características Transición del producto (Adaptabilidad de nuevos entornos): Transportabilidad, Reusabilidad.

Modelo BOEHM (1978)

Se caracteriza por centrarse en la calidad del producto final, agrega algunas características al modelo de McCall.

El modelo de Boehm tiene como finalidad que a través de la calidad del software, el mismo: (1) realice lo que desea el usuario, (2) utilice recursos informáticos de manera correcta y eficiente, (3) sea fácil de utilizar y aprender; y (4) sea bien diseñado, codificado, probado y mantenido. Este modelo es similar al de McCall ya que presenta una jerarquía de características, está basado en un amplio rango de características e incorpora 19 criterios que incluyen características de performance del hardware.

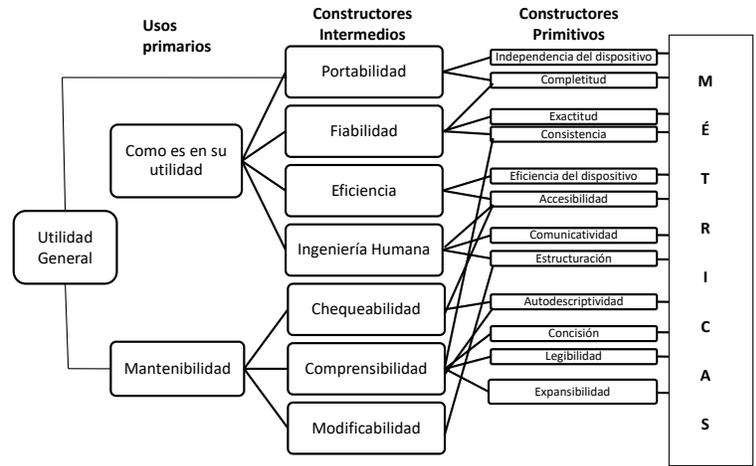


Figura 1. Modelo de Boehm divide la calidad en tres características. Fuente: (OLSINA, 2002)

Modelo GILB (1988)

El modelo de Gilb plantea la creación de una especificación de requisitos de calidad para cada proyecto que deben escribir conjuntamente el usuario y el analista. Es un modelo que permite determinar una lista de características que definen la calidad de la aplicación.

Puede ser de 2 tipos: (1) Originales y (2) de modelos tradicionales.

Las características se pueden medir mediante varias sub-características o métricas detalladas. Para cada una de ellas, se deben especificar los siguientes conceptos:

1. Nombre y definición de la característica.
2. Escala o unidades de medición.
3. Recopilación de datos o prueba.
4. Valor previsto.
5. Valor óptimo.
6. Valor en el sistema actual.
7. Comentarios.

Modelo GQM (Goal – Question – Metric):

Este modelo planteado por Basili y Weiss (1984) tiene la particularidad que las mediciones se hacen sobre objetivos que se trazan comenzando el desarrollo del proyecto. “GQM define un objetivo, refina éste objetivo

en preguntas y define métricas que intentan dar información para responder a éstas preguntas”.

En el siguiente gráfico se muestran las fases del método GQM que son en su orden la planificación, la definición, la recopilación de datos y la interpretación.

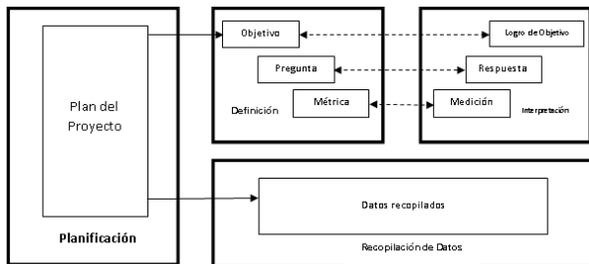


Figura 2. Metodología GQM. Fuente: *Medición y Estimación de software. Piattini (2003)*

Este modelo permite elegir las métricas que se relacionen a las metas más importantes de los problemas más urgentes.

Modelo FURPS

Modelo de calidad propuesto por Robert Grady y Hewlett Packard Co (HP) en 1987. Esta propuesta contempla, por un lado 5 características de las cuales se deriva su nombre (Funcionalidad, Facilidad de Uso, Confiabilidad, Performance y Facilidad de Soporte), y por otro, que los requisitos se clasifiquen en dos categorías: requisitos funcionales (F), que son los que especifican funciones que el sistema debe ser capaz de realizar sin tener en cuenta las restricciones físicas; y requerimientos no funcionales (URPS), que puntualizan atributos del sistema o del medioambiente del sistema.

MODELO	CATEGORÍA	CARACTERÍSTICA
FURPS	Requerimientos Funcionales (F)	Funcionalidad
	Requerimientos No Funcionales (URPS)	Usabilidad
		Confiabilidad
		Desempeño
	Facilidad de soporte	

Tabla 1. Características del modelo de FURPS.

MODELO DROMEY

El modelo DROMEY presentado por Robert Dromey(1995), aparece como una propuesta que permite construir y utilizar un modelo de calidad práctico para evaluar etapas de: determinación de los requerimientos, diseño e implementación.

Este modelo plantea la calidad del producto por medio de la definición de sub-características además que organiza tres modelos para cada etapa del proceso de desarrollo:

Modelo de requerimientos.

Modelo de diseño.

Modelo de implementación.

El modelo de calidad utiliza las características de: Eficiencia, Confiabilidad, Facilidad de Mantenimiento, Portabilidad, Facilidad de Uso y Funcionalidad. Y unas categorías que implican propiedades de calidad que son: Correctitud, internas, contextuales, y descriptivas, [8].

Los pasos para la aplicación del modelo de Dromey son los siguientes, [3]:

- Seleccionar el conjunto de atributos que se necesitan evaluar.
- Realizar una lista de todos los componentes o módulos del sistema.
- Identificar las propiedades de calidad de cada componente.
- Determinar cómo afecta cada propiedad en los atributos de calidad.
- Evaluar el modelo de calidad.

Modelo SATC (Software Assurance Technology Center)

Los modelos de calidad mencionados anteriormente no entran en el detalle de las métricas necesarias para el área de ingeniería de requisitos exceptuando el Dromey, al ser modelos que proporcionan un entorno de trabajo y unas directrices para crear el modelo de calidad de software adecuado a nuestras necesidades. En 1996 el [SATC \(Software Assurance Technology Center\) de la NASA](#) estableció su propio modelo de calidad de software, siguiendo el modelo ISO 9126.

En el año 1996 cuando el SATC decidió las métricas que iban a formar parte de su modelo de calidad para requisitos, tomó como criterio de decisión que las mediciones a realizar tuvieran un coste razonable en su ejecución, es decir, que la cantidad de tiempo y dinero invertido en obtener los datos no fuera desorbitado. Es por esa razón que el número de métricas no es grande y se dejaron fuera del modelo otra serie de factores que habrían ayudado a conseguir una mayor precisión en la calidad de las especificaciones de requisitos software de la NASA.

Este modelo utiliza un amplio rango de medidas o métricas y; tiene objetivos, atributos y métricas asociadas a los procesos de desarrollo y al software propiamente dicho. Este modelo define un conjunto de metas u objetivos relacionados al producto de software y atributos del proceso que permiten realizar indicaciones de la probabilidad de éxito de los objetivos, [3].

Satc posee cuatro metas u objetivos mencionados a continuación, éste material se ha tomado exactamente del proyecto de Fernanda Scalone [3]:

1- Calidad de los Requerimientos: El objetivo de esta meta es que los documentos de requerimientos estén completos, no ambiguos y entendibles.

Esta meta tiene los siguientes atributos:

Ambigüedad: Requerimientos con múltiples significados.

Integridad: Items a ser especificados.

Facilidad de entender: Documento legible.

Trazabilidad: Trazabilidad de los requerimientos generales respecto del código y de las pruebas.

2- Calidad del Producto: Un objetivo importante de un proyecto de desarrollo de software es desarrollar código y documentación que se correspondan con los requerimientos del proyecto. Esta meta u objetivo tiene los siguientes atributos:

Estructura / Arquitectura: La evaluación de un módulo para identificar posibles errores e indicar problemas potenciales en la facilidad de uso y facilidad de mantenimiento.

Reutilización: Utilizar el software en diferentes contextos o aplicaciones.

Facilidad de mantenimiento: Es el esfuerzo requerido para localizar y corregir un error en un programa.

Documentación: Tener la adecuada documentación del código interno y la documentación externa.

3- Efectividad de la implementación: El objetivo de la efectividad de la implementación es maximizar la efectividad de los recursos dentro de las actividades programadas en el proyecto. Los atributos de este objetivo son:

Uso del recurso: El uso del recurso relacionado a la etapa apropiada del proyecto.

Cumplimiento de los porcentajes: Avances realizados en los ítems.

4- Efectividad de la prueba: Los objetivos de la prueba de efectividad es ubicar y reparar las fallas del software. El atributo es la corrección. Una vez generado el código, se realizan pruebas de unidades, pruebas finales y pruebas de aceptación.

Metodología SQAE

Esta metodología desarrollada por el MITRE Corporation cuyo objetivo es producir un sistema de evaluación que satisfaga el objetivo de producir resultados confiables en todas las etapas del ciclo de vida del software.

“Es una metodología que permite cuantificar los riesgos asociados al software” [3]

La metodología SQAE se hace basado en modelos como McCall, Boehm y Dromey.

Se definen cuatro áreas de calidad en la SQAE: Mantenibilidad, Evolutibilidad, Portabilidad, Descriptibilidad. Cada una con sus propios factores de calidad que proveen una estructura para medir la calidad de un sistema, éstas son: consistencia, independencia, modularidad, documentación, auto-descripción, control de fallos, simplicidad del diseño.

Las áreas de calidad se usan para definir los conceptos de riesgos del ciclo de vida y se expresan como la suma de varios factores que abarcan aspectos del concepto a medir. Estas áreas son definidas por medio de atributos y porcentajes usados en el proceso de Evaluación, [3].

El uso de porcentajes en los factores de calidad deriva de las áreas de calidad. Por cada factor de calidad se tiene definido un mapeo entre el factor de calidad y una o más áreas de calidad. Cada factor de calidad está definido por un conjunto de atributos, los cuales contemplan distintas facetas del factor de calidad en cuestión [8].

ISO 9126

Se presenta la primera versión de este modelo en 1991, luego es presentada la ISO 9126:1 en el 2001 quedando como actualmente se conoce, la ISO 25000 luego es otra mejora a este modelo y que se presentará más adelante en este mismo capítulo. La ISO 9126 se descompone en dos partes: Modelo para la calidad Externa e Interna y Modelo para calidad en uso.

La calidad externa evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas. Esta calidad es medible en el comportamiento del producto. La calidad interna evalúa el total de atributos que un software debe satisfacer teniendo en cuenta condiciones especificadas. Esta calidad es medible a partir de las características intrínsecas [3].

La ISO 9126 descompone la calidad en seis factores que se mencionan a continuación:

Funcionalidad.

Confiabilidad.

Usabilidad.

Eficiencia.

Capacidad de mantenimiento.

Portabilidad.

El modelo se desglosa de la siguiente manera:

ISO 9126-1: Modelo de Calidad.

ISO 9126-2: Métricas Externas.

ISO 9126-3: Métricas Internas.

ISO 9126-4: Calidad en Uso.

Según ISO/IEC 9126-1 (2001), algunos atributos pueden contribuir a más de una sub característica y una característica puede ser influenciada por más de un atributo. Se puede presentar también que los niveles de ciertos atributos internos influyan en los niveles de algunos atributos externos. Por ejemplo, la característica de confiabilidad puede ser medida externamente mediante la observación del número de fallas en un período determinado de tiempo de ejecución durante una prueba del software, e internamente mediante la inspección de las especificaciones detalladas y el código fuente para evaluar el nivel de tolerancia a fallos; se dice que los atributos internos son indicadores de los atributos externos.

ISO/IEC 25000: SQUARE

Presentada en el 2005 la ISO/IEC se basa en una combinación de la ISO 9126 y en ISO 14598 (Evaluación del software). Este conjunto de normas deriva su nombre de Software Quality Requirements and Evaluation y tiene como objetivo fundamental guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad.

La ISO/IEC 25000 se centra más en el producto de software, es una mejora respecto a la 9126 presentando algunos cambios en las características y sub características de calidad interna/externa y de calidad en uso de la misma.

La ISO 25000 se divide en:

ISO/IEC 2500n – División de Gestión de Calidad [9].

Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la serie SQUARE.

ISO/IEC 2501n – División de Modelo de Calidad [9].

La norma de este apartado presenta un modelo de calidad detallada incluyendo características para calidad interna, externa y en uso.

ISO/IEC 2502n – División de Medición de Calidad [9].

Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación.

ISO/IEC 2503n – División de Requisitos de Calidad [9].

Estas normas ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación.

ISO/IEC 2504n – División de Evaluación de Calidad [9].

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para la evaluación de productos software.

CONCLUSIONES

Hoy en día se pueden encontrar diversos modelos para establecer la medición de software, las empresas desarrolladoras de software tienen la oportunidad de explorar los diferentes modelos y ver cuál es el que se ajusta a sus necesidades de implementación de calidad al interior de sus desarrollos.

Se puede analizar que entre el año 1977 y el 2005 las nuevas características y necesidades del cliente fuesen incrementando los diferentes modelos de medición de los productos para cualquier tipo de software.

Resta realizar otro documento que establezca los atributos, características y métricas principales de los modelos mencionados como también las formas de medición que han usado con los mismos. El tema de la calidad y la medición del software tienen bastante por explorar y esta búsqueda de modelos es sólo el inicio de posteriores trabajos que ayuden a los investigadores de tema en su labor.

REFERENCIAS

- [1] Z. Cataldi, «Facultad de Ingeniería, Universidad de Buenos Aires».
- [2] M. Piattini, F. O. García, J. Garzás y M. F. Genero, Medición y
- [3] F. Scalone, «Estudio comparativo de los modelos y estándares ingeniería-en-calidad.pdf. [Último acceso: 19 04 2014].
- [4] C. M. Calero, «Modelos de calidad.WQM, PQM, e-commerce, departamento.html. [Último acceso: 10 2014].
- [5] R. S. Pressman, Ingeniería del Software.Un enfoque práctico S
- [6] 1. 6.-1. -. I. S. C. D. A. C. o. I. S. C. G. (IEEE, «Universidad Nacio
- [7] J. McCall, P. Ricards y G. Walters, «Scientific Research,» 1977.
- [8] H. Solano y I. Torres, «Análisis de frameworks para el desarrollo <http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source> [Último acceso: 11 07 2014].
- [9] M. G. Piattini, F. García y I. Caballero, Calidad de Sistemas Info
- [10] S. M. Abrahão y O. Pastor, «Calidad de sistemas Web,» de Ca
- [11] D. Reifer, «Web Development: Estimating Quick-to-Market So
- [12] M. G. Piattini y F. O. García, Calidad en el desarrollo y manten