

EXPLORACIÓN DE MODELOS Y ESTÁNDARES DE CALIDAD PARA EL PRODUCTO SOFTWARE

JORGE JAIR MORENO

*Magister en Informática
Docente de Ingeniería de Sistemas
Universidad del Cauca
jjmoreno@unicauca.edu.co*

LILIAM PAOLA BOLAÑOS

*Estudiante Ingeniería de Sistemas
Grupo GTI
Universidad del Cauca
lpbolanos@unicauca.edu.co*

MANUEL ALEJANDRO NAVIA

*Estudiante Ingeniería de Sistemas
Grupo GTI
Universidad del Cauca
mnavia@unicauca.edu.co*

*Fecha de Recibido: 15/12/2009
Fecha de Aprobación: 26/05/2010*

RESUMEN

La importancia de la calidad de software en la disciplina de la Ingeniería del software es ampliamente reconocida en la actualidad, sin embargo desde el punto de vista de los modelos y estándares hacia el producto, el desarrollo de estos durante décadas, la sobreabundancia de información, el alto costo y el acceso limitado a esta información, impiden un acercamiento de estos a los ingenieros de software en pro de la calidad del producto software [1] al interior de la organización. Para mitigar lo anterior, éste documento tiene como propósito hacer una exploración de los modelos de calidad de producto software (McCall [2], Boehm [3], FURPS [4], ISO 9126 [5], Dromey [6], SQA E [7], Bansiya [8] y Quint2 [9]), donde se describe la perspectiva de calidad que cada uno plantea y su relación con otros modelos. Posteriormente se dan a conocer los atributos de calidad que cada modelo plantea. Por último y como resultado de la reflexión de los autores, se proponen algunos elementos esenciales, en relación a la calidad del producto software.

PALABRAS CLAVES: Modelos de calidad, Estándares, Producto software, Atributos

ABSTRACT

Related with software quality, since the point of view about models and standards toward product, their development for almost decades, the over abundance of information, the high cost and the limited access to this information, hamper an approach toward the software engineers in order to the software product quality [1] inside the organization. To mitigate before, this document has a propose to realize a brief explanation of software product quality models (McCall [2], Boehm [3], FURPS [4], ISO 9126 [5], Dromey [6], SQA E [7], Bansiya [8] y Quint2 [9]), in that the quality perspective of each and the relationship with other models are described. Later, the quality attributes from each model are raised. Finally, a trial to clear up some essential elements is tried, related with the software product quality.

KEYWORDS: Quality models, Standard, Software product, Attributes

1. INTRODUCCIÓN

La calidad del producto software ha sido un tema que han abordado un sin número de autores en diferentes modelos [10], [11], [12], refiriéndose a ella como el conjunto de atributos deseables que posee un producto software, los cuales son medibles (cuantitativa o cualitativamente), permitiendo hacer comparaciones para conocer si se cumple con las expectativas del cliente o no. Sin embargo, la calidad es un concepto cuya acepción con bastante frecuencia depende del ángulo desde donde se le mire. En ese sentido, desde la perspectiva de las empresas, la calidad está directamente relacionada con la competitividad y madurez en los productos y procesos, implicando mayores y mejores oportunidades tanto para su crecimiento como expansión al mismo tiempo que se intenta alcanzar la satisfacción de sus clientes [13]; de otro lado está la mirada del cliente, para quien es importante contar con una empresa que garantice la calidad y confiabilidad [13] de sus productos, de tal forma que puedan alcanzar mayor competitividad en su propio contexto de negocio.

En este trabajo, se intentó dar una mirada a los modelos sobre calidad del producto con el objeto de clarificar cómo se ha construido el concepto a lo largo del tiempo mediante varios aportes. En ese mismo orden de ideas, existieron varias razones que motivaron la elegibilidad de los modelos de calidad que se tratan en el presente trabajo. La primera de ellas consiste en su orientación preponderante hacia la evaluación del producto más que hacia el proceso, la segunda se relaciona con la estructura que presenta cada modelo donde algunos de los atributos encontrados ocasionalmente presentan denominaciones y/o definiciones similares que deben conciliarse; por último, la frecuencia de aparición de los atributos en cada modelo, que se infiere en ocasiones de forma mecánica (conteo), pero en otras, requiere de la conciliación mencionada con anterioridad. En este mismo sentido, los modelos McCall [2], Boehm [3], FURPS [4], ISO 9126 [5], Dromey [6], SQA [7], Bansiya [8] y Quint2 [9], los cuales aparecen referenciados en una apreciable cantidad de trabajos [14], [15], fueron elegidos para los propósitos de este trabajo.

De otro lado, el agregado de información recabada a partir de estos modelos, sirvió como producto intermedio que favorecería el alcance de varios resultados finales del componente de investigación en el trabajo de grado MACMHA, cuyo objeto más allá del cumplimiento de

sus propios objetivos consiste en impulsar la creación de una línea de investigación en la Universidad del Cauca, que motive a docentes y estudiantes a profundizar en la temática de métricas, heurísticas y patrones, relacionados con mejores prácticas en el diseño arquitectónico del producto software.

Por lo general, los modelos de calidad describen el “qué” mas no el “cómo” de la valoración de calidad [16] de un producto software, en consecuencia, existe una sensación de vacío en el conocimiento que causa problemas y frustración a aquél que intenta aplicarlos. De aquí, es evidente la interpretación particular que cada modelo le da al concepto de calidad, ocasionalmente abruma y confunde al usuario de aquellos modelos, impidiéndole usarlos efectiva y eficazmente.

Con el objeto de responder a las anteriores preocupaciones, el proyecto MACMHA busca en un principio mejorar la comprensión de los modelos de calidad, facilitando el acercamiento entre los ingenieros de software novatos y el conocimiento relacionado con la valoración del producto software, de tal forma que se favorezca el uso de mejores prácticas que propendan por la calidad del producto software. Otro problema, lo constituye el hecho relacionado con una aparente divergencia y/o ambigüedad entre los modelos al momento de escoger y definir los atributos de calidad que plantea el autor de cada modelo, lo anterior plantea un serio inconveniente para las personas inexpertas en el tema, pues a partir de estas divergencias/ambigüedades aparentes, se generan importantes confusiones que paralizan procesos de mejora por parte de los interesados. Agregado a lo anterior, existe una proliferación de modelos de este tipo que si bien ofrece alternativas suficientes para casi todos los gustos, al mismo tiempo crean una sobresaturación de la información disponible [17] que con mucha frecuencia no es la oficial o tiene serias restricciones de acceso, dejando al interesado en un mar de incertidumbres que lo regresan a la mencionada parálisis. Por último, la persona intenta hacer uso de las buenas prácticas [18] en busca de la valoración de calidad del producto software (interna o externa), podría desistir y continuar usando prácticas inadecuadas, inmaduras y caóticas, perpetuando así el imaginario colectivo de la comunidad de desarrollo en relación con la calidad del software. En éste documento, se intenta hacer un acercamiento y esclarecimiento de los criterios de calidad [19] relacionados con el producto software que coinciden en los diferentes modelos considerados y que fueron mencionados anteriormente.

Este documento está estructurado como sigue: inicialmente se hace una breve descripción de los modelos de calidad de software elegidos y cuya preponderancia se inclina hacia el producto software; luego se da a conocer una apreciación sobre cómo algunos modelos han influido sobre otros; posteriormente se presentan algunos atributos de calidad que se manifiestan de forma concurrente en los modelos, y finalmente se muestran algunas conclusiones.

2. DESARROLLO

2.1 Calidad de Software a partir de Modelos

Los modelos y estándares de calidad de software recopilados han sido tema de discusión para gran cantidad de autores [14], dada la trascendencia del tema y el gran despliegue que ha tenido a lo largo de los años. Los modelos aquí presentados recogen amplitud de perspectivas que diversos autores toman con respecto a la valoración de calidad. Aunque hay muchos más, se tratan éstos en particular porque son los de mayor acogida en gran parte de los autores consultados [15] y que cumplen con los criterios de elegibilidad requeridos para el estudio de MACHMA.

Con el fin de centrar algunos conceptos de calidad que tengan relación entre sí, y a su vez pertenezcan a la clasificación de modelos o estándares de calidad del producto software, se hace necesario tener en cuenta algunos criterios de selección que sirvan para elección de aquellos modelos que pueden ser de interés para este trabajo.

- C1: *Disponibilidad*: grado en que es posible acceder a la información existente. Se refiere a la facilidad de obtener la información.
 - 1: la información no se encuentra disponible al público en general
 - 2: Hay disponibilidad de algunos documentos pero es limitado el acceso.
 - 3: Se encuentra información suficiente disponible para ser usada.
- C2: *Claridad*: Grado en que el modelo es presentado y si posee mecanismos explicativos sobre su uso. Se refiere a que tan sencillo puede ser entender el modelo, influyen factores como: estructura, idioma y presentación del modelo.
 - 1: El modelo no es claro o se dificulta su entendimiento, no posee mecanismos de ayuda sobre el modo de emplearlo.

2: El modelo es presentado en forma clara, sin embargo no posee mecanismos de ayuda sobre el modo de emplearlo.

3: El modelo es presentado en forma clara, posee mecanismos explicativos acerca de su modo de empleo.

- C3: *Adaptabilidad*: Grado en el que el modelo posee la capacidad de adaptarse a distintas situaciones dependiendo del producto al que se va aplicar.

1: El modelo no es adaptable. Se presenta de forma rígida para su uso.

2: El modelo puede ser adaptado pero exige ciertas reglas a seguir.

3: El modelo permite ser adaptado.

- C4: *Complejidad*: Grado en el que el modelo describe todas sus partes en su totalidad sin dejar por fuera información importante. Un modelo completo se considera que posee descripción de atributos, métricas y mecanismos de ayuda para llegar a la medición.

1: El modelo no menciona toda la información necesaria. Se encuentra incompleto

2: El modelo describe medianamente sus componentes, sin embargo deja algunos elementos por fuera. Está incompleto.

3: El modelo describe todas sus partes. Está completo.

Adicional a los 4 primeros criterios de selección surge un 5 que es de interés para este trabajo y tiene relación a si es o no un modelo para producto, este se describe a continuación.

- C5: *Área de aplicación*: aplicabilidad del modelo a las diferentes áreas de calidad del software.

1: Modelo de proceso, metodología o estándar (no incluye modelo)

2: Puede ser modelo de proceso y producto al mismo tiempo.

3: Modelo para producto software

Para cada modelo o estándar existe la posibilidad en que no exista el caso en el que no aplica el criterio en este caso se usa la notación NA, la cual no suma puntuación.

En la tabla 1 se muestra los resultados del estudio realizado.

Tabla 1. Modelos y Estándares de calidad de software

Modelo / Estándar	Criterios					Total
	C1	C2	C3	C4	C5	
McCall	2	2	2	2	3	11
Boehm	2	2	2	2	3	11
FURPS	1	2	3	1	3	10
GILB	2	NA	NA	NA	3	5
IEEE 610.12	2	NA	NA	NA	1	3
SATC	2	2	2	3	2	11
Dromey	1	2	2	2	3	10
ISO 14598	1	1	NA	NA	2	4
SQAE	2	2	2	2	3	11
Bansiya	2	2	2	2	3	11
GQM (Goal / Question / Metric)	2	1	3	1	2	9
IEEE 1061	2	NA	NA	NA	1	3
ISO 9126	2	2	2	2	3	11
QUINT2	1	2	2	2	3	10
PQM (Portal Quality Model)	2	1	3	1	2	9
Six sigma	2	NA	NA	NA	1	3
ISO 9000-3 (TickIT)	2	NA	2	NA	1	5
ISO 15504 (SPICE)	2	2	1	NA	1	6
ISO 12207 o IEEE / EIA 12207	2	2	2	NA	1	7
Personal Software Process (PSP)	1	NA	NA	NA	1	2
Practical Software Measurement (PSM)	1	NA	NA	NA	1	2
Bootstrap	1	NA	NA	NA	1	2
Team Software Process (TSP)	1	NA	NA	NA	1	2
CMMI (Capability Maturity Model Integration)	2	3	1	1	1	8
ISO 90003	2	NA	NA	NA	1	3
ISO 25000 (SQUARE)	2	2	2	NA	1	7
ISO 20000	2	NA	NA	NA	1	3

2.2 Modelos de Calidad

Los modelos y estándares de calidad que se muestran a continuación, presentan propuestas o perspectivas frente a cómo valorar la calidad del producto, además se

ha tratado con cierta profundidad con respecto a otros [20][21].

2.2.1 El modelo de McCall - 1977

El modelo fue escrito por McCall [2], [10], Richards y Walters, siendo publicado en el año 1977 en un documento denominado “*Factors in software quality*”. El modelo refleja perspectivas del desarrollador y del usuario, además presenta una estructura jerárquica para organizar los factores divididos en tres aspectos de calidad de software (revisión, transición y operación), como se muestra en la figura 1.

Los factores de calidad planteados por McCall se miden a través de 21 criterios o métricas de calidad que él propone; el problema es que dichos criterios se calculan a través de preguntas dicotómicas del tipo “SI”-“NO”, las cuales son contestadas por una o varias personas, lo cual podría implicar subjetividad dado que cada una puede evaluar la calidad de forma diferente.

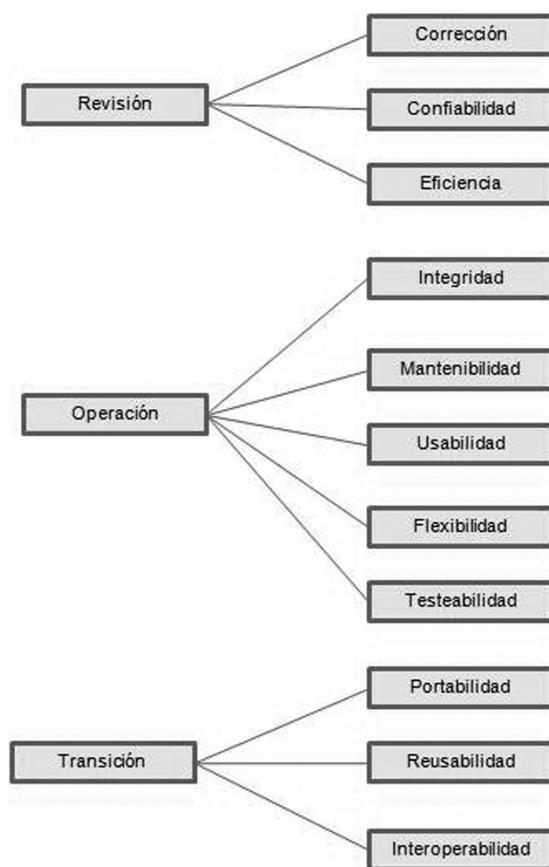


Figura 1. Factores de Calidad de McCall

2.2.2 El modelo de Boehm – 1978

El segundo modelo presentado en este documento es el propuesto por Barry Boehm en 1978 [3]. Éste define la calidad de software en términos de atributos cualitativos

y los mide usando métricas. El modelo no es muy distinto al de McCall, porque muchos de sus factores de calidad son los mismos. Éste modelo también presenta sus factores de calidad estructurados jerárquicamente de alto a bajo nivel como se muestran en la Tabla 2.

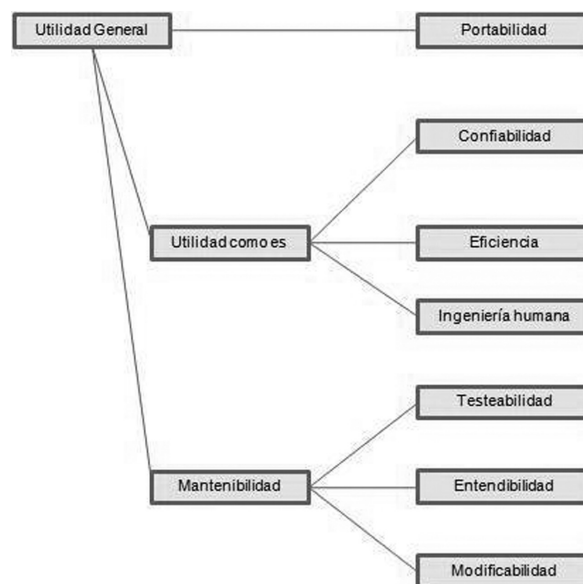


Figura 2. Factores de Calidad de Boehm

2.2.3 El modelo FURPS – 1987

En 1987 Hewlett-Packard desarrolló una serie de factores de calidad que reciben el acrónimo de FURPS [4], que incluye cinco (5) categorías principales por sus nombres en inglés: Funcionalidad (Functionality), Usabilidad (Usability), Confiabilidad (Reliability), Desempeño (Performance) y Soportabilidad (Supportability), de aquí el nombre del modelo. Su estructura se puede visualizar en la figura 2.

2.2.4 El modelo ISO 9126 - (1991/2001)

El estándar ISO 9126 presenta su primera versión en 1991, luego en 2001 es remplazado por ISO 9126:1 [5] que además cuenta con tres ítems adicionales para ayudar a la mejora de la calidad del producto software (Métricas externas, Métricas internas, Métricas de calidad en uso). Además presenta una estrecha relación con el estándar ISO 14598:1. El estándar ISO-9126 define un modelo, basado en modelos ya existentes como McCall, Boehm y US Air Force.

El estándar ISO 9126 presenta dos partes, el *Modelo de calidad para calidad externa e interna*, y el *Modelo de calidad para calidad en uso*. La primera parte del

modelo, en las figuras 3 y 4, especifica 6 características de calidad externa, las cuales están divididas en sub-características que representan la calidad interna y

tienen influencia sobre las características externas. La segunda, en la Tabla 4, presenta cuatro características de calidad, para ser evaluadas desde la vista del usuario.

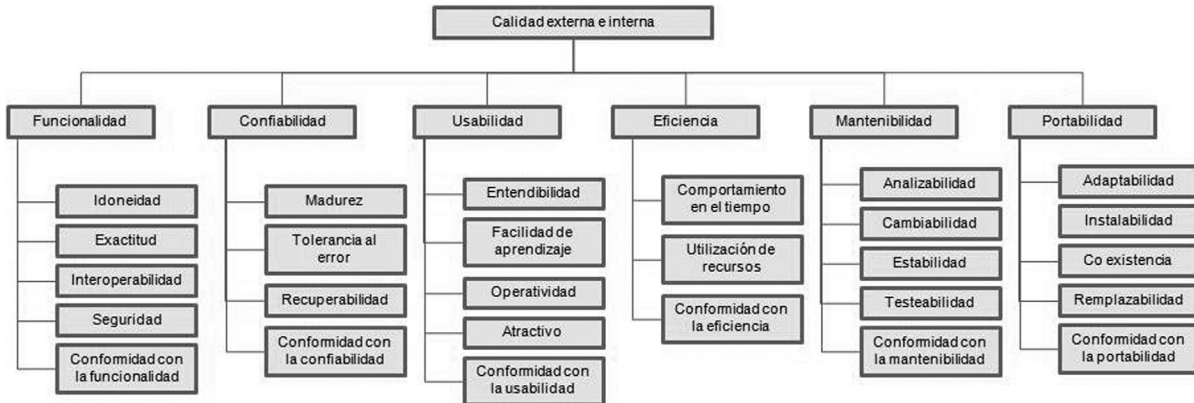


Figura 3. Modelo de calidad para calidad Interna y Externa



Figura 4. Modelo de calidad para calidad en Uso

2.2.5 El modelo de Dromey – 1995

El modelo de Dromey [6] presenta un esquema de 6 relaciones binarias entre 3 entidades definidas (Conjunto de componentes, propiedades que acarrear calidad de los componentes, atributos de calidad de alto nivel) en la Figura 5, cuatro de las cuales permiten evaluar la calidad desde la perspectiva del producto o el proceso (aquellas de la punta de flecha sombreada).

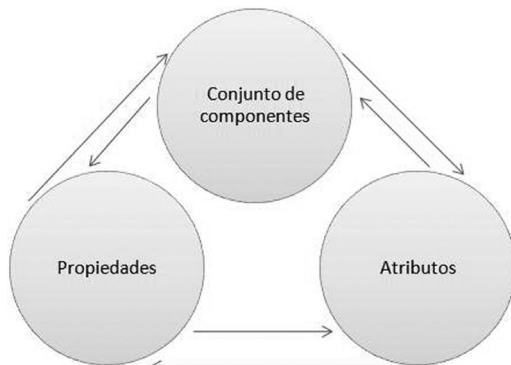


Figura 5. Esquema general del modelo de Dromey

Éste modelo se propone como alternativa al inconveniente que se presenta cuando los atributos de alto nivel no pueden ser medidos directamente sobre el software. En respuesta a esto, los atributos de alto nivel se pueden obtener con la construcción de componentes que representen un conjunto de propiedades del producto, señalando aquellas que afectan los atributos de calidad, como: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad, Portabilidad y Reusabilidad.

2.2.6 El modelo SQAE – 1995

Robert A. Martin and Lawrence H. Shafer (MITRE) crearon el SQAE (Software Quality Assessment Exercise) [7] para proveer una serie de herramientas y métodos de evaluación que den una medida de calidad de software que sea repetible y consistente, además de asociarle el riesgo. El aseguramiento de calidad que provee SQAE se enfoca en el riesgo relacionado con diferentes áreas de calidad y produce una lista de riesgos conducidos y elementos mitigables que pueden ayudar para hacer elecciones juiciosas cuando se seleccionan desarrolladores y/o mantenedores de software.

SQAE se ha basado en modelos tales como: Boehm, McCall y Dromey, además del estándar ISO/IEC 9126 (desarrollado paralelamente). Las cuatro áreas de calidad con las que SQAE trabaja son: Mantenibilidad,

Evolución, Portabilidad y Consistencia, además presenta siete factores (Independencia, Modularidad, Documentación, Auto descripción, Control anomalía, Diseño simple) para medir la calidad. Figura 6

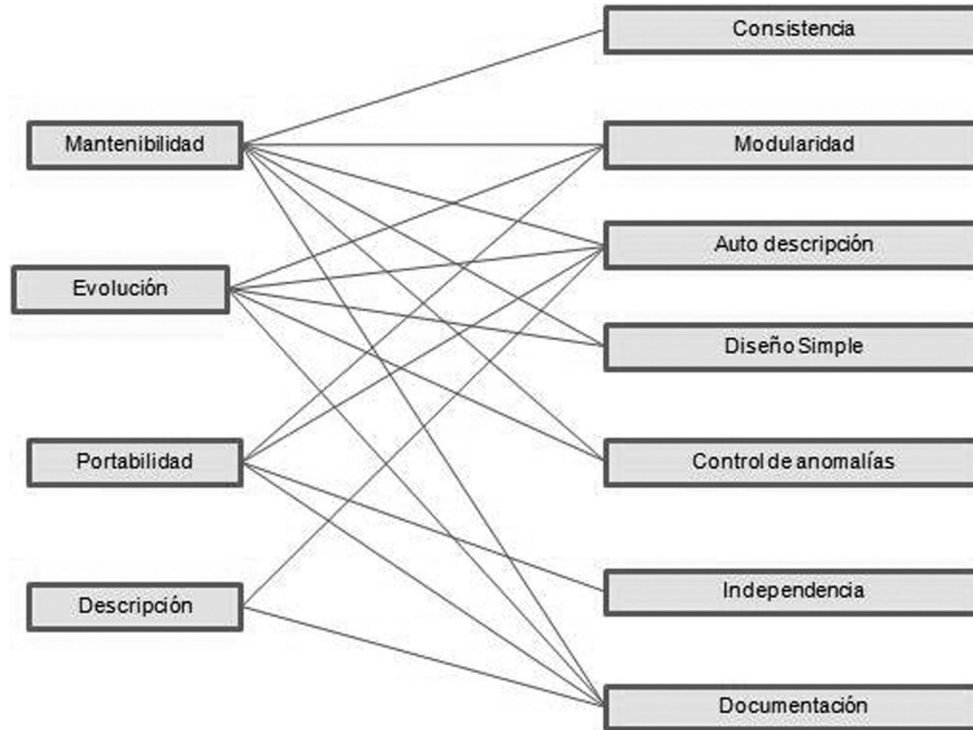


Figura 6. Modelo SQAE

2.2.7 El modelo de Bansiya – 1997

El modelo denominado QMOOD+ (Quality Model for Object-Oriented Design) fue desarrollado por Jagdish Bansiya y Carl G. Davis [8]; basado en otros como McCall, Dromey, e ISO 9126. Este framework es una metodología para el desarrollo de modelos de calidad de estilo Bottom-up (de abajo hacia arriba), proporcionando un acercamiento en el cual se asegure

que los detalles de low-level (bajo nivel) sean bien especificados y computables. Como se aprecia en la Figura 7, consta de cuatro niveles (L1 a L4) conectados entre sí, además contiene 6 atributos de calidad (Reusabilidad, flexibilidad, comprensibilidad, funcionalidad, extensibilidad y efectividad) y 11 propiedades de diseño que complementan a los atributos. Además posee una serie de métricas para la evaluación de los mismos.



Figura 7. Niveles y enlaces del modelo de Bansiya (QMOOD)

2.2.8 El modelo QUINT2 -2002

Este modelo se autodenomina “modelo de calidad de software ISO extendido” [9], dado que es un súper-conjunto del grupo de características y sub-características

del modelo ISO 9126. Las sub-características están dadas para cada una de las características. La estructura general del modelo se muestra en la Figura 8:

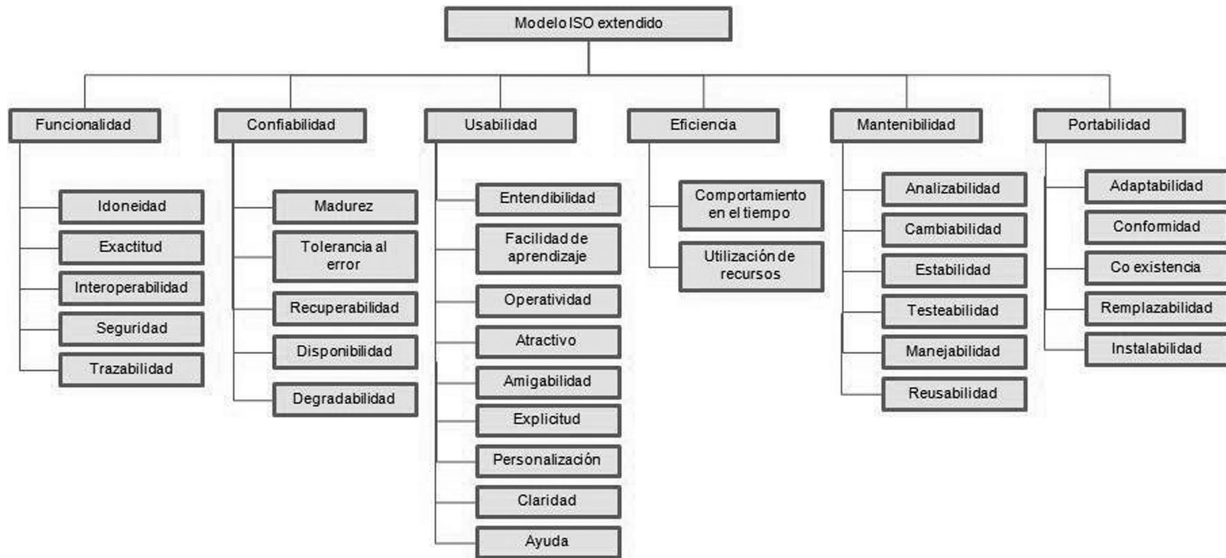


Figura 8. Atributos de Quint2. Características y sub-características

2.3 Importancia del Estudio

La literatura revisada anteriormente cubre varios aspectos de los modelos de calidad de software en general, los más conocidos (en orden cronológico de aparición, sección 2) con el fin de extraer elementos relevantes, no redundantes ni ambiguos que permitan construir un criterio más unificado sobre cada concepto de calidad a partir de diversos autores [22]. En este sentido, el estudio realizado justifica su importancia esencial en los siguientes aspectos:

- **Primero.** El estudio intenta establecer condiciones iniciales que sean favorables para el acercamiento del conocimiento a la comunidad académica y empresarial (Pymes) con el objeto que las personas emprendan caminos de mejoramiento viables que involucren mejores prácticas encaminadas a la búsqueda de la calidad en el producto de software.

- **Segundo.** Mitigación de las divergencias (aparentes o reales) entre los modelos de calidad elegidos, mediante la conciliación de la definición y el sentido de los atributos de calidad propuestos por cada modelo, buscando así, el asentamiento de un marco de referencia para la base de conocimiento del proyecto MACMHA.

- **Tercero.** Mejora en las condiciones de disponibilidad y accesibilidad de la información sobre calidad del producto software a la comunidad de software.

2.4 Relación entre Modelos

Por lo general, cada modelo propuesto presenta algunas dependencias con los anteriores y al mismo tiempo influye en sus contemporáneos y sucesores. Para ilustrar esta situación en la evolución de los modelos de calidad del producto software, se propone un diagrama que ilustra las relaciones de descendencia entre ellos, véase la Figura 9:

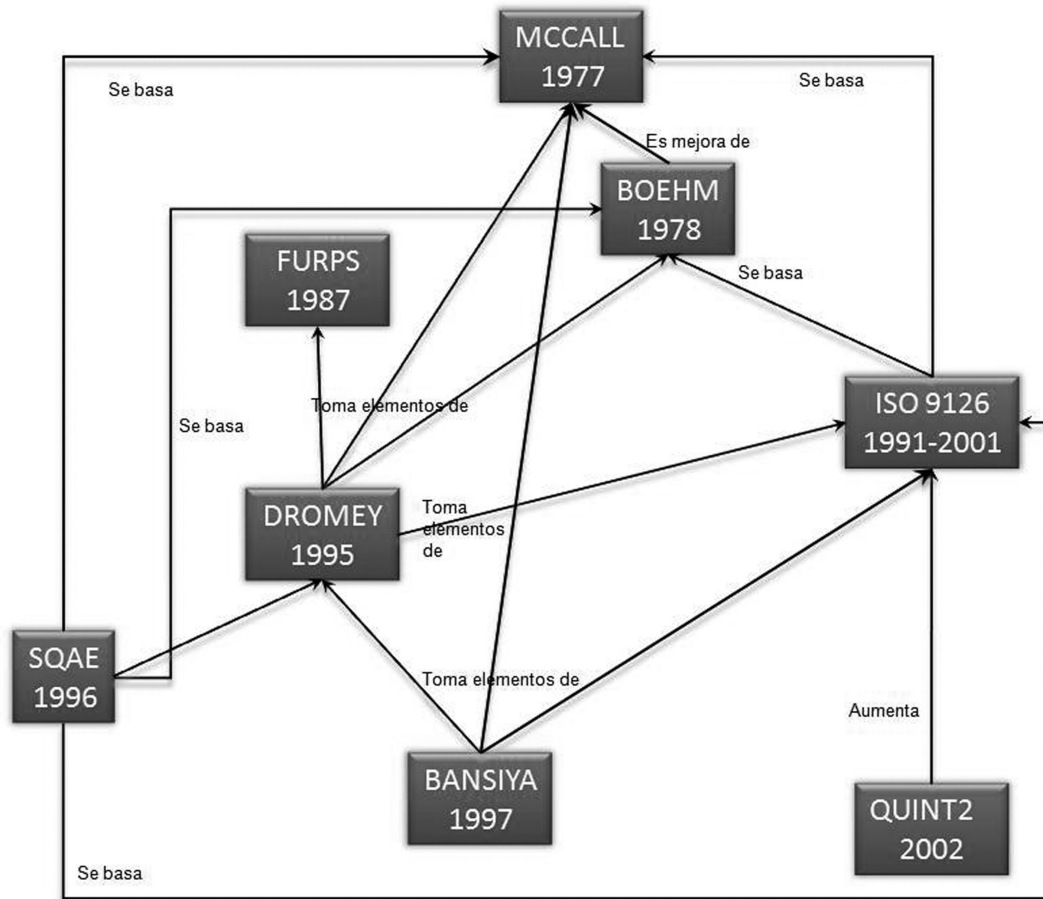


Figura 9. Relación entre modelos

Como se muestra, los modelos se han basado en anteriores para presentar su percepción de la calidad del producto software. Por ejemplo en el caso de SQAE, tiene sus orígenes en ISO 9126 en su versión del año 1991, pero fue desarrollado paralelamente con la versión que se presentó en 2001, mientras que Bansiya por su parte se fundamenta en los modelos de McCall, Dromey e ISO-9126. De otro lado, nótese cómo el modelo FURPS ha sido inspiración de alguna manera para el modelo de Dromey, al igual que otros anteriores, como el modelo de McCall y Boehm.

Parece evidenciarse, según la bibliografía, que existe un antecedente entre un modelo y otro. Las relaciones de dependencia mutua existentes entre los modelos surge a partir de sus elementos componentes (atributos de calidad), los cuales son propuestos en cada modelo. Aún cuando algunos modelos nombran los atributos de manera diferente, en esencia puede tratarse del mismo concepto, además se tuvo en consideración el

modo en que pueden ser aplicados los atributos en cada modelo, y como consecuencia evidenciar si conservan sus cualidades, convirtiéndose en elementos básicos desde los que se puede realizar medición de calidad de software. La gráfica surge a partir de la recopilación de los modelos que poseen la mayor cantidad de referencias en diversos trabajos [14][15]. Por lo tanto, las relaciones aquí mostradas están sujetas a cambios según la aparición o consideración de otros modelos no tenidos en cuenta hasta el momento.

2.5 Atributos de Calidad

Respecto a los modelos de calidad de software, se puede percibir que cada autor maneja una notación diferente para referirse a un mismo concepto. McCall [21] utiliza factores y criterios; mientras que Boehm [21] utiliza características y primitivas; para FURPS [21] son factores y atributos; de otro modo, ISO 9126 [21] maneja atributos o características y sub-atributos de

calidad, y para Dromey [21] son atributos y propiedades de calidad de software. Todos estos conceptos se refieren a clasificaciones que cada autor hace para su modelo, en definitiva a cómo va a estar dada la medida

de calidad. Para llegar a una clasificación ordenada de dichos conceptos, apoyándonos en Meyer [17], se llegó a la siguiente unificación para los términos aquí utilizados, en la Tabla 2:

Tabla 2. Selección de conceptos por modelo

	McCall	Boehm	FURPS	ISO 9126	Dromey	Bansiya
Concepto	Factor	Característica Primitiva	Factor	Atributo ó Característica	Atributo	Atributo de calidad
	Criterio	Característica	Atributo	Sub - Atributo	Propiedad	Propiedad de diseño
	Métrica	Métrica	-	Métrica	Métrica	Métrica

Cabe anotar que, basado en la sección 2.1, la forma de los conceptos definidos en el modelo SQA es igual a la usada en el estándar ISO 9126:1, razón por la que no se muestra en la tabla anterior.

En la Tabla 2 se puede apreciar cómo las notaciones de las palabras para cada modelo se refieren a un mismo concepto; en la primera línea se encuentran diferentes definiciones para lo que al final se trata de llamar factor externo, los cuales pueden ser percibidos por los usuarios de un software, y pueden ser medidos por los factores internos (presentados en la segunda fila de los conceptos), que no son percibidos directamente por el usuario sino por personas en las organizaciones desarrolladoras del software que se encargan de su diseño e implementación. Algo en lo que al parecer los autores concuerdan consiste en denominar métricas a la forma en la que se calcula o se hace la medida cuantitativa para satisfacer el cumplimiento del factor interno [17]. Para este trabajo se decidió tomar la conceptualización de Atributo (Factor externo), Sub-Atributo (Factor interno) y Métrica, entendido que el Atributo es aquél que se encuentra en un nivel más alto y por tanto su medida es externa, el Sub-Atributo se encuentra en un nivel intermedio y su medida es interna y finalmente, la Métrica como la expresión (matemática, cuantitativa o

cualitativa) que permite la valoración directa del Sub-atributo en cuestión.

La naturaleza de un atributo está determinada por la caracterización del aspecto de calidad, que busca determinar en el software a valorar, esto es, si un atributo se puede valorar “a simple vista” sin necesidad de acceder al código fuente con miras a “mejorar” la característica que en él se pretende observar. Así, atributos tales como la usabilidad y la eficiencia pueden ser evaluados a través de criterios que pueden ser expresados con solamente el análisis visual de su desempeño. Por lo tanto, se puede determinar que los atributos pueden ser de naturaleza externa ó interna según la característica software que se pretende evaluar.

En este documento se procura mostrar una comparación de los modelos de calidad del producto software en relación a los atributos y sub atributos que aparecen en cada uno, con el fin de seleccionar aquellos que tenga mayor coincidencia, importancia o influencia sobre otros atributos de calidad. Por lo tanto, se muestran a continuación, en la Tabla 3, los modelos elegidos previamente desde los cuales se extraen los atributos con mayor coincidencia en aparición dentro de los modelos:

Tabla 3. Modelos vs. Atributo

MODELO/ESTANDAR vs ATRIBUTOS	M1	M2	M3	M4	M5	M6	M7	M8	Frecuencia atributo
Mantenibilidad	x	x	x	x	x		x	x	7
Testabilidad	x	x	x				x	x	5
Portabilidad	x	x		x	x		x	x	6

MODELO/ESTANDAR vs ATRIBUTOS	M1	M2	M3	M4	M5	M6	M7	M8	Frecuencia atributo
Reusabilidad	x			x		x		x	4
Exactitud	x	x	x				x	x	5
Confiabilidad	x	x	x	x			x	x	6
Eficiencia	x	x	x	x			x	x	6
Usabilidad	x		x	x			x	x	5
Comprensibilidad		x				x	x	x	4
Consistencia	x	x	x	x	x				5
Auto descripción	x	x		x	x				4
Funcionalidad			x	x		x	x	x	5

Convenciones utilizadas: M1: McCall, M2: Boehm, M3: FURPS, M4: Dromey, M5: SQAE, M6: Bansiya, M7: ISO 9126, M8: Quint 2

Los atributos mostrados en la tabla anterior, inicialmente han sido escogidos, debido a su mayor coincidencia de aparición en los diferentes modelos aquí comentados, lo cual expresa la posibilidad de que exista consistencia en la forma de medir presentada por los modelos de producto, además puede ser un indicio de la influencia que aparenta tener los atributos de bajo nivel sobre los atributos de alto nivel; por otro lado, se ha tenido en cuenta su definición en los distintos modelos vistos, que en algunos casos tiende a ser similar. Un aspecto a considerar es que no siempre un atributo posee el mismo nombre en cada uno de los modelos, para esta situación se intenta obtener la definición más clara, y se toma el nombre más común.

A continuación se muestra una clasificación más detallada, en la Figura 10 y 11, la cual es una refinación de un trabajo realizado con antelación [1] y la información reunida durante la elaboración del presente. En éste se pretende mostrar una organización de los atributos y sub-atributos que los modelos aquí mencionados poseen, presentada en capas para su mejor comprensión:

Capa1. En esta capa se han organizado los factores de calidad que son independientes entre sí. El atributo se considera como externo.

Capa2. Esta capa indica todos los factores de calidad cuya valoración contribuye a su vez a la valoración de los factores de alto nivel que están en la capa 1.

Capa3. Considerados como los atributos internos, que tienen influencia sobre los externos.

Compartido: Aquél que le sirve a más de un atributo externo.

Exclusivo: Aquél que sólo le sirve a un atributo externo o interno únicamente.



Figura 10. Capas

En este punto aún no se ha incluido la capa de métricas como lo indica [1], sin embargo se ha ampliado la gama de atributos y sub atributos, pertenecientes a los modelos de Dromey y Bansiya dado que tratan la calidad desde el código fuente con orientación a objetos, conectándose así a modelos anteriormente tratados como ISO-9126 y FURPS, para obtener una mejor visión de lo que puede ser la calidad para el producto software tanto a nivel de valoración de usuario como de valoración de código fuente, y así conseguir una conexión más confiable desde el código fuente hasta la vista de usuario.

3. CONCLUSIONES

Hoy en día se encuentran diversos modelos de calidad para el producto software, en donde cada uno de ellos presenta conceptos acerca de una serie de atributos de calidad, que pueden diferir de otros modelos. Esto puede generar confusión al lector que esté interesado en la calidad de su producto software y en consecuencia podría aplicar mal las técnicas de búsqueda/valoración calidad, además de prestarse para múltiples interpretaciones. Este trabajo presenta una perspectiva de la calidad tomando en cuenta los atributos que cada modelo posee, buscando con esto que cualquier persona interesada en la calidad del producto software, pueda acercarse a ella, de forma que se disponga de un material de fácil acceso e interpretación, además de mostrar sencillez y claridad para realizar una valoración del producto software.

Los modelos de calidad para producto software, en su mayoría, se han inspirado en modelos anteriores, lo cual hace que tengan una gran posibilidad de encontrar divergencias y similitudes entre sus atributos de calidad, además de contemplar mejoras que los modelos anteriores no han tomado en cuenta. Es decir, los modelos más actuales para producto software, se han basado en algunos ya existentes buscando especializaciones que permitan mejorar inconsistencias o refinar diversidad de aspectos mostrados en ellos.

En el proyecto MACMHA se espera construir iterativamente un colectivo de los atributos conciliados a partir de los modelos considerados, sus métricas relacionadas y las heurísticas (recomendaciones de buena voluntad) de calidad que los favorecen para el producto software orientado a objetos. La razón por la cual se tomó como concepto base la palabra atributo de calidad, se debió a la simplicidad y homogeneidad semántica del concepto mismo, permitiendo su relación y valoración a la luz de sus métricas y heurísticas aliadas. Además, variedad de autores toman éste criterio de manera similar y no ambigua, situación que impulsa aún más a adoptarlo.

La selección de atributos tiene por objeto identificar los aspectos básicos que permiten evaluar la calidad de un producto software, dada su naturaleza y la

posibilidad de uso que muestran para analizar un producto software. La labor realizada aquí pretende enriquecer el conocimiento en el área de calidad para el producto software a partir de los atributos, también se busca llegar a personas interesadas en el tema o que estén en la misma línea de investigación de una forma sencilla y clara, además que en los entornos académicos o empresariales sirva de apoyo para la valoración del producto software.

En la elaboración de este trabajo se presentaron los siguientes inconvenientes encontrados por los autores: (1) las ambigüedades entre las definiciones y nombres de los atributos presentados en cada modelo, (2) los atributos que presentan definiciones confusas e incompletas, (3) la confusión causada por errores de traducción, por último (4) el desafío de darle consistencia al modelo propuesto, mediante la concepción de criterios para la ubicación adecuada de algunos atributos de calidad.

Mientras exista el distanciamiento entre los modelos y los usuarios de los mismos, será difícil lograr un avance positivo hacia un cambio en la postura desde un desarrollo caótico hacia uno centrado en mejores prácticas que propendan por el mejoramiento continuo de la calidad del producto software. En este sentido el proyecto MACHMA ha percibido esta problemática y ha trazado un plan de esfuerzos sucesivos para mitigar esta situación.

De otro lado, MACHMA ofrecerá un allanamiento de los baches presentados el conocimiento residente en la bibliografía disponible. Este proceso incluirá la búsqueda y determinación tanto de las métricas que permiten la valoración de cada atributo de calidad contemplado, como las heurísticas que contribuyen a alcanzarlo.

Finalmente, MACHMA, en una fase más tardía se propone encontrar aquellas relaciones de dependencia cuya naturaleza implique influencias (positivas, negativas) entre atributos/sub-atributos de calidad, con el objeto de permitir a los arquitectos de software “balancear” los intereses de un producto software en términos de la calidad esperada por sus clientes finales.

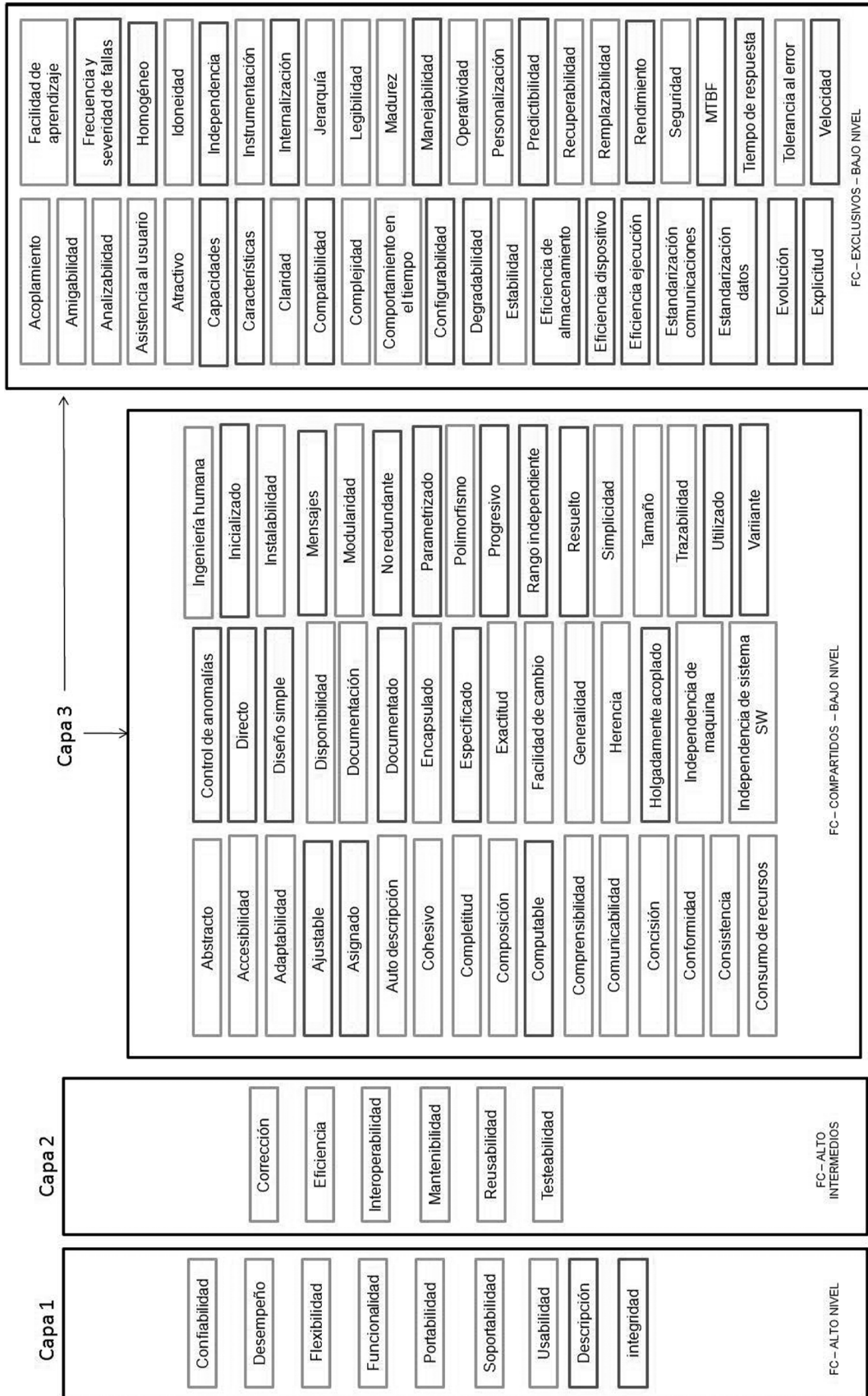


Figura 11. Atributos por capa

4. REFERENCIAS

- [1] J. Moreno, H. Andrade, L. Bolaños, “Compilación de un Modelo para Evaluar Atributos de Calidad en Productos Software”. Revista Enlace Informático No 1, diciembre de 2007, pp. 99 – 111.
- [2] J. McCall, P. Richards, G. Walters, “Factors in software quality”. New York: The National Technical Information Service, 1977. 42 p.
- [3] B. Boehm, “Characteristics of Software Quality”. New York: North-Holland Publishing, 1978. 130 p.
- [4] R. Grady, D. Caswell, Software Metrics: Establishing a Company-Wide Program. Mountain View, California: Prentice Hall, 1987. 275 p.
- [5] THE INTERNATIONAL STANDARD INSTITUTE. ISO/IEC 9126-1, Information Technology – Software Product Quality – Part 1: Quality Model. Geneva, Suiza: The ISO Publisher, 2002. 32 p.
- [6] G. Dromey, “A Model for Software Product Quality”. IEEE Transactions on software engineering No 21. Griffith, Australia, Software Quality Institute, febrero de 1994. pp. 146 – 162.
- [7] J. Bansiya, C. Davis, “A Hierarchical Model for Quality Assessment of Object-Oriented Designs”. Huntsville, 1997, 214 p. Doctoral dissertations. The University of Alabama in Huntsville.
- [8] B. Van Zeist, et al. “Kwaliteit van software-producten”. Ervaringen met een praktijkmodel. Kluwer Bedrijfswetenschappe. Deventer: The Netherlands. 1996. 113 p. (En Holandés)
- [9] M. A. Côté, W. Suryn, “Evolving a Corporate Software Quality Assessment Exercise: A Migration Path to ISO/IEC 9126”. Software Quality Professional Journal No 6. Chicago, Illinois: The SQP Journal. Mayo de 2004. pp. 4 – 17.
- [10] R. Pressman. “Ingeniería de Software, un enfoque práctico”. New York: McGraw - Hill. 5° Edition, 2002. 601 p.
- [11] ICONTEC Instituto Colombiano de Normas Técnicas y Certificación. Sistema de Gestión de la Calidad: Fundamentos y Vocabulario (NTC-ISO 9000). Bogotá: División de publicidad. 2000. 37 p.
- [12] Jetter, A. Assessing Software Quality Attributes, with source code metrics. Zurich, 2006. 56 p. Diploma Thesis on Informatics. University of Zurich.
- [13] Fernandez, V. “Marco de Referencia Centrado en la Arquitectura para la Mejora de Características de Usabilidad en el Desarrollo de Aplicaciones Web Construidas por MIPYMEs”. [Trabajo de grado]. Facultad de Ingeniería Electrónica. Universidad Del Cauca. 2008, 170 p.
- [14] Scalone, F. “Estudio Comparativo De Los Modelos Y Estándares De Calidad Del Software”. [Trabajo de maestría]. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires. 2006, 461 p.
- [15] Calderón, S. Modelos De Calidad Enfocados A La Usabilidad, Aplicados En Los Procesos De Desarrollo De Sistemas De Información. Ciudad Real, España. Universidad De Castilla-La Mancha, Escuela Politécnica Superior. Available: <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Shlomi%20Calderon%20-%20Modelos%20de%20Calidad%20Enfocados%20a%20la%20Usabilidad%20-%20Doc.pdf>. [citado 12 de abril de 2008].
- [16] Rubio, D. Un Proceso Definido Para La Planeación Temprana De Proyectos Software. La Plata, Argentina, 2005, 113 p. Tesis de Maestría en Ingeniería de Software. Universidad Nacional De La Plata, Argentina.
- [17] Meyer, B. Object – Oriented Software Construction, Second Edition. Santa Barbara, California: Prentice-Hall, 1991. 1370 p.
- [18] The IEEE Institute. IEEE 1061-1998, Standard for a Software Quality, Metrics Methodology. New York: The IEEE Publisher. 1998. 26 p.
- [19] The IEEE Institute. IEEE 610.12 -1990, IEEE Standard Glossary of Software Engineering Terminology. New York: The IEEE Publisher. 1990. 84 p.
- [20] Gilb, T. Principles of Software Engineering Management. Boston, USA: Addison Wesley, 1987. 464 p.
- [21] Olmedilla, J. Revisión Sistemática de Métricas de

Diseño Orientado a Objetos. Revista UPM, No 2. Madrid, España: Universidad Politécnica de Madrid, Septiembre de 2005. 12 p.

- [22] Moreno, J. Bolaños, L. Navia, M. Informe: Encuesta para el diagnóstico respecto a prácticas del producto software. Informe de Investigación. Universidad del Cauca. Popayán, 2008, 13 p.