

CALIDAD EN MODELOS CONCEPTUALES: UN ANÁLISIS MULTIDIMENSIONAL DE MODELOS CUANTITATIVOS BASADOS EN LA ISO 9126

Beatriz Marín, Nelly Condori-Fernández, Oscar Pastor

Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.

Camino de Vera s/n. 46022 Valencia, España

[_bmarin, nelly, opastor}@dsic.upv.es](mailto:{bmarin,nelly,opastor}@dsic.upv.es)

Resumen

La ISO 9126 [3] es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que son medidos por una serie de métricas. Estas métricas miden artefactos obtenidos en etapas tardías del desarrollo de software, aumentando el costo de detección y corrección de errores. Por esta razón, en la literatura ha surgido un mayor interés por la definición de métricas que pretenden evaluar una o varias de las características de calidad definidas en el estándar ISO 9126, en etapas tempranas del desarrollo de software. En el presente artículo se revisa un conjunto representativo de métricas para medir modelos conceptuales y se identifican las sub-características de calidad a las que contribuye cada métrica. Además, se presentan los resultados de un análisis llevado a cabo sobre esas métricas, desde las perspectivas de calidad definidas por Krogstie [9], como son la calidad semántica, semántica percibida, sintáctica, pragmática, de lenguaje, de conocimiento, física y social. A partir de estos resultados, se identifican las razones por las que dichas métricas no aseguran la calidad de los productos de software desde una visión industrial.

1. INTRODUCCIÓN

La producción de software de manera automática o semi-automática está fuertemente posicionada en la industria de desarrollo de sistemas de software [16]. La adopción de esta tecnología implica la realización de modelos conceptuales, que posteriormente son utilizados como entrada para el proceso de generación de código.

De este modo, los modelos conceptuales constituyen el recurso fundamental de los procesos de generación de software en entornos MDA (Model Driven Architecture) [20], y por este motivo, es deseable asegurar la calidad de estos modelos, evitando la propagación de los errores y el alto coste de su corrección. En este sentido, una gran cantidad de investigadores ([10], [19], [10], [14], [8], [21], [15], [17], [7], [13], [2]) ha definido modelos cuantitativos de calidad que pre-

tenden contribuir al aseguramiento de la calidad de los modelos conceptuales.

Dado el auge de los métodos orientados a objetos sobre los métodos estructurados, estos modelos cuantitativos de calidad están orientados a medir primitivas de los modelos conceptuales orientados a objetos, es decir, ponen énfasis en conceptos como el encapsulamiento, la herencia, el polimorfismo y la complejidad de las clases, ya sea en modelos estáticos o en modelos dinámicos.

En este artículo se revisan exhaustivamente estos trabajos, y se identifican las sub-características de calidad definidas en el estándar ISO 9126 que tienen relacionadas, teniendo como objetivos analizar estas propuestas según las perspectivas de calidad definidas por Krogstie, y posteriormente identificar las razones por las que estos modelos cuantitativos de calidad no aseguran la calidad de los productos de software deseada por las organizaciones.

Este artículo está organizado de la siguiente manera: la sección 2 describe el modelo de calidad propuesto por Krogstie y el estándar ISO 9126. La sección 3 presenta las propuestas más significativas asociadas a la medición cuantitativa de modelos conceptuales y las sub-características de calidad a las que contribuyen con su medición. En la sección 4 se realiza un análisis de las características definidas en el estándar ISO 9126 y de las propuestas presentadas en la sección 3 según las perspectivas de Krogstie. La sección 5 presenta las razones por las cuales dichas métricas no aseguran la calidad de los modelos conceptuales desde una visión industrial. Finalmente, la sección 6 presenta las conclusiones y los trabajos futuros.

2. MODELOS DE CALIDAD

Dado que este artículo se centra en analizar las métricas que contribuyen a las características y sub-características de calidad definidas en el estándar ISO 9126 [3] según las perspectivas de Krogstie [9], a continuación se introducen estos modelos de calidad para facilitar el entendimiento de las secciones posteriores.

2.1 Modelo de calidad de Krogstie

Krogstie ha propuesto un modelo que conceptualiza la calidad de los sistemas de información teniendo en cuenta la evolución continua y las características sociales de las organizaciones.

Tal como se puede ver en la Figura 1, el modelo de Krogstie propone seis conjuntos, que al relacionarse entre sí dan lugar a los diferentes tipos de calidad:

- **Audiencia:** Este conjunto no se refleja explícitamente en la figura, ya que no define tipos de calidad al relacionarse con los demás conjuntos del modelo. Este conjunto corresponde a los actores individuales, organizacionales y

técnicos que evalúan la calidad del modelo.

- **Extensión del lenguaje:** Corresponde a los hechos que se pueden realizar con la gramática y vocabulario del lenguaje de modelado utilizado.
- **Dominio de modelado:** Corresponde a los hechos que se pueden realizar en una situación específica.
- **Externalización del modelo:** Corresponde a los hechos de un modelo particular, que refleja parte de la realidad percibida, y que está escrito en un lenguaje particular.
- **Conocimiento de los participantes:** Corresponde al conocimiento explícito y relevante que tiene la audiencia.
- **Interpretación de la audiencia:** Corresponde a los hechos que la audiencia piensa que se reflejan en la externalización del modelo.

Las correspondencias entre los hechos de los diferentes conjuntos antes mencionados definen ocho tipos de calidad: semántica, semántica percibida, sintáctica, pragmática, de lenguaje, de conocimiento, física y social.

La *calidad física* consiste en la externalización del conocimiento de algún actor social por el uso de un lenguaje de modelado conceptual, de manera que la externalización del modelo sea persistente y esté disponible a la audiencia.

La *calidad sintáctica* es la correspondencia entre la externalización del modelo y la extensión del lenguaje en que el modelo está escrito.

La *calidad semántica* es la correspondencia entre el modelo y el dominio, es decir, que el modelo refleje el dominio. Para esto, se revisa la validación y la completitud del modelo. La validación de la calidad semántica consiste en que todos los hechos del modelo sean correctos y relevantes al dominio. La completitud consiste en que el modelo

contiene todos los hechos correctos y relevantes para el dominio.

La *calidad semántica percibida* es la correspondencia entre las interpretaciones de los participantes de un modelo y su conocimiento actual del dominio de ese modelo.

La *calidad pragmática* es la correspondencia entre el modelo y la interpretación de la audiencia.

La *calidad social* corresponde al acuerdo entre las interpretaciones de los participantes.

La *calidad de lenguaje* consiste en la habilidad del lenguaje de modelado para capturar el dominio, la facilidad de aprender, usar y entender el lenguaje de modelado por los participantes, la formalización del lenguaje para permitir su ejecución, y la relevancia que tiene el conocimiento del dominio sobre el lenguaje.

La *calidad de conocimiento* es la cantidad de conocimiento que tienen los participantes del dominio.

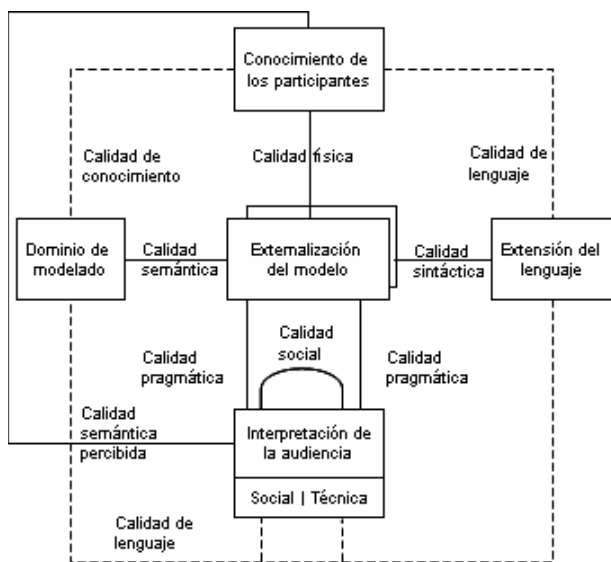


Figura 1. Modelo de calidad de Krogstie [9]

De esta manera, aplicando el modelo de calidad de Krogstie a los modelos conceptuales, se puede identificar la calidad de esos mode-

los mediante la conjunción de los ocho tipos de calidad antes definidos.

2.2 Estándar ISO/IEC 9126

En 1991, la Organización Internacional de Estándares (ISO) en conjunto con la Comisión Electrotécnica Internacional (IEC) propusieron un estándar para la evaluación de la calidad del software, denominado ISO 9126. En el 2001, este estándar fue reemplazado por dos estándares relacionados: el estándar ISO/IEC 9126, que especifica características y métricas de la calidad del software; y el estándar ISO/IEC 14598, que especifica la evaluación de productos de software. La forma en que se relacionan estos estándares se muestra en la Figura 2.

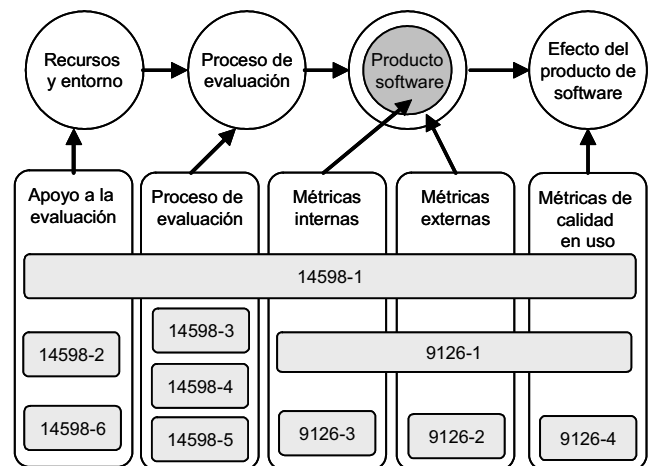


Figura 2. Relación entre ISO 9126 e ISO 14598

El estándar ISO/IEC 9126 se compone de cuatro partes: modelo de calidad [3], métricas externas [4], métricas internas [5] y métricas para la calidad en uso [6].

En la primera parte, se describen detalladamente seis características y sub-características de calidad para los productos de software (Tabla 1). De esta manera, se puede determinar el grado de calidad estos productos según la evaluación de estas características y sub-características.

Tabla 1. Características y sub-características de calidad definidas en el estándar ISO 9126 [3]

Característica	Sub-características
Funcionalidad	Adecuación Corrección Interoperabilidad Seguridad Conformidad
Fiabilidad	Madurez Tolerancia a fallos Recuperabilidad Conformidad
Usabilidad	Comprensibilidad Aprendibilidad Operabilidad Atractividad Conformidad
Eficiencia	Comportamiento temporal Utilización de recursos Conformidad
Mantenibilidad	Analizabilidad Cambiabilidad Estabilidad Facilidad de prueba Conformidad
Portabilidad	Adaptabilidad Instalabilidad Coexistencia Reemplazabilidad Conformidad

En la segunda, tercera y cuarta parte de la ISO 9126 se describen métricas y atributos para evaluar la calidad del software. Estas métricas y atributos están enfocadas a medir artefactos obtenidos en fases tardías del ciclo de desarrollo de software, complicando la detección y corrección de los problemas de las etapas iniciales, que se propagan a las etapas posteriores.

Para evitar esta situación, han surgido varias propuestas que contribuyen al aseguramiento de la calidad en etapas tempranas del ciclo de desarrollo de software, que se presentan en la siguiente sección.

3. METRICAS PARA MODELOS CONCEPTUALES ORIENTADOS A OBJETOS

La mayoría de los trabajos revisados, no han identificado la correspondencia de las métricas que proponen con las características y sub-características de calidad definidas en el estándar ISO 9126. Para identificarla, se han revisado las métricas presentes en las partes [4], [5] y [6] del estándar, y se han analizado las correspondencias que existen con las métricas definidas por los diversos autores, mediante la abstracción de los conceptos esenciales que mide cada métrica. Una vez identificadas estas correspondencias, se han identificado las respectivas sub-características y la característica de calidad a la que contribuye cada una de las métricas.

A continuación se presentan las propuestas de métricas revisadas y el resultado de la identificación de las respectivas correspondencias:

- Métricas CK [10] [19]: Estas métricas fueron definidas para medir la complejidad, el acoplamiento, la cohesión, la herencia y la comunicación inter-clases del diseño orientado a objetos. Estas métricas ayudan a asegurar la característica de mantenibilidad de los productos de software, ya que dependiendo de esas medidas se puede predecir si el modelo y su posterior implementación serán fáciles de corregir, mejorar o adaptar a nuevos requisitos. La Tabla 2 muestra las relaciones entre las métricas CK y las sub-características de la mantenibilidad:

Tabla 2. Relación entre métricas CK y sub-características ISO 9126

Métrica	Descripción	Sub-característica
WMC	Métodos ponderados por clase, según su complejidad.	Analizabilidad
DIT	Longitud máxima desde una clase hasta la raíz de la jerarquía de herencia en la que participa.	Analizabilidad Cambiabilidad Estabilidad
NOC	Número de descendientes de una clase.	Analizabilidad Cambiabilidad
CBO	Número de clases con la que una clase está acoplada.	Analizabilidad Cambiabilidad Estabilidad
LCOM	Número de métodos que acceden a uno o más de los mismos atributos.	Analizabilidad Cambiabilidad Estabilidad
RFC	Número de métodos existentes en el conjunto de respuestas de una clase.	Analizabilidad

- Métricas de Li y Henry [21]: Estas métricas están enfocadas a apoyar la medición de la característica de mantenibilidad del software. Para esto, se focalizan en mediciones de primitivas del diseño del software. La Tabla 3 muestra las métricas de Li y Henry, y las sub-características que tienen relacionadas:

Tabla 3. Relación entre métricas de Li y Henry, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
MPC	Número de métodos invocados en una clase.	Analizabilidad
DAC	Número de atributos en una clase, que tienen como tipo otra clase.	Analizabilidad Cambiabilidad Estabilidad
SIZE1	Variación de la métrica LOC – número de líneas de código – definida inicialmente para el lenguaje ADA.	Analizabilidad
SIZE2	Número de atributos más número de métodos locales.	Analizabilidad

- Métricas MOOD [2]: Estas métricas fueron definidas para medir el uso de los mecanismos de diseño orientado a objetos a nivel de sistema, tales como la encapsulación, la herencia, el polimorfismo y el paso de mensajes. Estos mecanismos de diseño orientado a objetos pretenden apoyar el aseguramiento de la característica de mantenibilidad del software. La Tabla 4 muestra las métricas MOOD y las sub-características de mantenibilidad con las que están relacionadas:

Tabla 4. Relación entre métricas MOOD y sub-características ISO 9126

Métrica	Descripción	Sub-característica
MHF	Proporción entre la suma de las invisibilidades de los métodos de todas las clases y el número total de métodos definidos en el sistema.	Analizabilidad
AHF	Proporción entre la suma de las invisibilidades de los atributos de todas las clases y el número total de atributos definidos en el sistema.	Analizabilidad
MIF	Proporción entre la suma de todos los métodos heredados de todas las clases y el número total de métodos en todas las clases.	Analizabilidad
AIF	Proporción entre el número de atributos heredados y el número total de atributos.	Analizabilidad
PF	Proporción entre el número real de diferentes situaciones polimórficas posibles para una clase y el máximo número de situaciones polimórficas posibles en la clase.	Analizabilidad Cambiabilidad

CF	Proporción entre el máximo número de acoplamientos posibles en el sistema y el número real de acoplamientos no atribuibles a herencia.	Analizabilidad Cambiabilidad Estabilidad
----	--	--

- Métricas de Lorenz y Kidd [14]: Estas métricas fueron definidas para medir las características estáticas del diseño del software, tales como el tamaño, el uso de herencia y el número de responsabilidades de una clase. De esta manera, estas métricas contribuyen a asegurar la mantenibilidad de los productos de software. La Tabla 5 muestra las métricas de Lorenz y Kidd y las sub-características a las que están relacionadas:

Tabla 5. Relación entre métricas de Lorenz y Kidd, y sub-características ISO 9126

Métrica	Descripción	Subcategoría
PIM	Número total de métodos públicos de una clase, que están disponibles como servicios para otras clases.	Analizabilidad Estabilidad
NIM	Número total de métodos de las instancias de una clase, ya sean públicos, privados o protegidos.	Analizabilidad
NIV	Número total de variables (privadas y protegidas) a nivel de instancia que tiene una clase.	Analizabilidad
NCM	Número total de métodos globales de una clase, es decir, métodos visibles por todas las instancias de la clase.	Analizabilidad Estabilidad
NVV	Número total de variables globales de una clase, es decir, visibles por todas las instancias de la clase.	Analizabilidad Estabilidad
NMO	Número de métodos sobrecargados en una subclase.	Analizabilidad Cambiabilidad Estabilidad
NHO	Número de métodos que hereda una clase.	Analizabilidad Cambiabilidad Estabilidad
NMA	Número total de métodos que se definen en una subclase.	Analizabilidad

APPM	Número medio de parámetros por operación.	Analizabilidad
SIX	Índice de especialización de una clase.	Analizabilidad Cambiabilidad
LOC	Número de líneas de código por método.	Analizabilidad
NOM	Número de mensajes enviados por un método.	Analizabilidad

- Métricas de Marchesi [15]: Estas métricas tienen como objetivo medir la complejidad del sistema, mediante el balanceo de responsabilidades entre paquetes y clases; y la cohesión y el acoplamiento entre entidades del sistema. Así, estas métricas ayudan a asegurar la característica de mantenibilidad definida en la ISO 9126. La Tabla 6 presenta las métricas de Marchesi sus sub-características relacionadas:

Tabla 6. Relación entre métricas de Marchesi, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
CL1	Peso total de las responsabilidades de una clase, ya sea heredadas o no.	Analizabilidad
CL2	Peso total de las dependencias de una clase.	Analizabilidad Cambiabilidad Estabilidad
PK1	Número de dependencias entre clases pertenecientes a un paquete.	Analizabilidad Cambiabilidad Estabilidad
PK2	Número de dependencias entre clases servidoras, pertenecientes a un paquete.	Analizabilidad Cambiabilidad Estabilidad
PK3	Promedio de los valores PK1 de un sistema.	Analizabilidad
OA1	Número total de clases del sistema.	Analizabilidad
OA2	Número total de jerarquías de herencia del sistema.	Analizabilidad
OA3	Promedio de los pesos de las clases.	Analizabilidad
OA4	Desviación estándar del peso de las clases.	Analizabilidad

OA5	Promedio de los números de dependencias directas de las clases del sistema.	Analizabilidad Cambiabilidad
OA6	Desviación estándar del número de dependencias directas de las clases del sistema.	Analizabilidad Cambiabilidad
OA7	Porcentaje de responsabilidades heredadas con respecto al total de responsabilidades de las clases del sistema.	Analizabilidad Cambiabilidad Estabilidad

- Métrica de Harrison et al [17]: Esta métrica mide el acoplamiento de una clase, y así pretende contribuir a asegurar la mantenibilidad de los productos de software. La Tabla 7 presenta la relación entre esta métrica y las sub-características definidas en el estándar:

Tabla 7. Relación entre métrica de Harrison et al, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
NAS	Número de asociaciones de una clase.	Analizabilidad Cambiabilidad Estabilidad

- Métricas de Bansiya & Davis [8]: Estas métricas fueron definidas para evaluar propiedades de diseño, como son la encapsulación, acoplamiento, cohesión, composición y herencia. A través de estas propiedades, estas métricas pretenden ayudar a asegurar la característica de mantenibilidad del software. La Tabla 8 presenta las métricas de Bansiya & Davis y las sub-características a las que están relacionadas:

Tabla 8. Relación entre métricas de Bansiya & Davis, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
DSC	Número total de clases en el diseño.	Analizabilidad
NOH	Número de jerarquías de clases.	Analizabilidad
ANA	Número medio de ancestros.	Analizabilidad Cambiabilidad
DAM	Relación entre el número de atributos privados y el total de atributos de una clase.	Analizabilidad Cambiabilidad Estabilidad
DCC	Número de clases diferentes con las que una clase está relacionada.	Analizabilidad Cambiabilidad Estabilidad
CAM	Relación entre los métodos de una clase, basándose en la lista de parámetros de los métodos.	Analizabilidad Cambiabilidad
MOA	Extensión de la relación de agregación, mediante el uso de atributos.	Analizabilidad Cambiabilidad Estabilidad
MFA	Relación entre número de métodos heredados por una clase y el número total de métodos que pueden ser accedidos por método de la clase.	Analizabilidad Cambiabilidad Estabilidad
NPM	Número de métodos que pueden mostrar comportamiento polimórfico.	Analizabilidad
CIS	Número de métodos públicos de una clase.	Analizabilidad Cambiabilidad Estabilidad
NOM	Número de métodos definidos en una clase.	Analizabilidad

- Métricas de Genero et al [10]: Estas métricas fueron definidas para medir la complejidad de los diagramas de clases, según diferentes tipos de relaciones (asociaciones, agregaciones, dependencias y generalizaciones). En [12], estos autores identifican que contribuyen a la evaluación de la mantenibilidad de los diagramas de clases, sin embargo tienen en cuenta la comprensibilidad como una sub-característica de la mantenibilidad.

Dado que la comprensibilidad no es una sub-característica de la mantenibilidad según el estándar ISO 9126, se ha analizado nuevamente cada una de las métricas presentadas para identificar las relaciones que efectivamente tengan con las sub-características de la mantenibilidad definidas en el estándar ISO 9126. La Tabla 9 muestra las métricas de Genero et al y las sub-características de la mantenibilidad que tienen relacionadas:

Tabla 9. Relación entre métricas de Genero et al, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
NASSOC	Número total de asociaciones dentro de un modelo de clases.	Analizabilidad
NAGG	Número total de relaciones de agregación dentro de un modelo de clases.	Analizabilidad
NDEP	Número total de relaciones de dependencia dentro de un modelo de clases.	Analizabilidad
NGEN	Número total de relaciones de generalización de un modelo de clases.	Analizabilidad
NGENH	Número de jerarquías de generalización en un modelo de clases.	Analizabilidad
NAGGH	Número de jerarquías de agregación en un modelo de clases.	Analizabilidad
MAXDIT	Máximo de los valores DIT obtenidos de cada clase del modelo de clases.	Analizabilidad
MAXHAGG	Máximo de los valores HAGG de cada clase del modelo de clases.	Analizabilidad

NASSOSC	Número de asociaciones por clase.	Analizabilidad Cambiabilidad Estabilidad
HAGG	Longitud de la ruta más larga desde la clase a las hojas dentro de una jerarquía de agregación.	Analizabilidad Cambiabilidad Estabilidad
NODP	Número de partes directas que contiene una clase que pertenece a una jerarquía de agregación.	Analizabilidad Cambiabilidad Estabilidad
NP	Número de partes de una clase Todo, ya sean directas o indirectas.	Analizabilidad Cambiabilidad Estabilidad
NW	Número de clases Todo de una clase parte.	Analizabilidad Cambiabilidad Estabilidad
MAGG	Número de clases Todo directas que tiene una clase en una jerarquía de agregación.	Analizabilidad Cambiabilidad Estabilidad
NDEPIN	Número de clases que dependen de una clase.	Analizabilidad Cambiabilidad
NDEPOUT	Número de clases de las que depende una clase.	Analizabilidad Cambiabilidad

- Métricas de JA Cruz-Lemus et al [7]: Estas métricas están enfocadas a medir los aspectos dinámicos de los modelos, específicamente toman como artefacto del diagrama de estados. Estos autores identifican que sus métricas contribuyen a asegurar la característica de mantenibilidad de los productos de software, según la sub-característica de comprensibilidad [11]. Debido a que la comprensibilidad es una sub-característica de la usabilidad en el estándar ISO 9126, se han identificado nuevamente las sub-características de la mantenibilidad relacionadas a dichas métricas. La Tabla 10 muestra las métricas de JA Cruz-Lemus et al y las

sub-características a las están relacionadas:

Tabla 10. Relación entre métricas de JA Cruz-Lemus et al, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
NEntryA	Número de acciones de entrada.	Analizabilidad
NExitA	Número de acciones de salida.	Analizabilidad
NA	Número de actividades.	Analizabilidad
NSS	Número de estados simples.	Analizabilidad
NCS	Número de estados compuestos.	Analizabilidad Cambiabilidad
NE	Número de eventos.	Analizabilidad
NG	Número de guardas.	Analizabilidad
NT	Número de transacciones.	Analizabilidad
McCabe	Número ciclomático de McCabe.	Analizabilidad

- Métricas de Kiewkanya et al [13]: Estas métricas están enfocadas a medir las interacciones de las clases, tomando como artefacto el diagrama de secuencia. Mediante las interacciones entre las clases, estas métricas pretenden ayudar a la medición de la mantenibilidad de los productos de software. Las métricas de Kiewkanya et al y las sub-características que tienen relacionadas son (Tabla 11):

Tabla 11. Relación entre métricas de Kiewkanya et al, y sub-características ISO 9126

Métrica	Descripción	Sub-característica
NOS	Número de escenarios.	Analizabilidad
WMBO	Peso de los mensajes entre objetos.	Analizabilidad
ANRM	Número promedio de los mensajes de retorno.	Analizabilidad Cambiabilidad

ANDM	Número promedio de los mensajes enviados.	Analizabilidad Cambiabilidad
ANET	Número promedio de los elementos en el cierre transitivo de los mensajes enviados directamente.	Analizabilidad
ANCM	Número promedio de mensajes de condición.	Analizabilidad

A pesar de que se ha encontrado un conjunto amplio y representativo de propuestas, que han definido modelos cuantitativos de calidad para que sean aplicados en modelos conceptuales orientados a objetos, se ha identificado que todos ellos apuntan a medir sólo una de las características definidas en el estándar ISO 9126: la mantenibilidad.

En la siguiente sección se presenta un análisis de las características definidas en el estándar ISO 9126 y de las métricas que apuntan a evaluar esas características, según las perspectivas de calidad definidas por Krogstie.

4. ANÁLISIS MULTIDIMENSIONAL

El análisis según las perspectivas de Krogstie se ha realizado en dos niveles. En primer nivel se han analizado las características definidas el estándar ISO 9126, y en segundo nivel se han analizado las métricas presentadas en la sección 3.

4.1 Análisis de ISO 9126

El análisis de las características de calidad definidas en la ISO 9126 según las perspectivas de Krogstie se realizará teniendo en cuenta a los modelos conceptuales orientados a objetos como productos de software. La *característica de funcionalidad* corresponde a la capacidad de los productos de software de proveer las funciones que satisfacen las necesidades de un dominio en particular. Esta característica en los modelos

conceptuales implica que el modelo tiene todas las funciones necesarias para que puedan ocurrir los hechos del dominio. Por esta razón, la funcionalidad se corresponde con la calidad semántica definida por Krogstie, que define que el modelo sea correcto y completo para los hechos que ocurren en un dominio específico.

La *característica de fiabilidad* consiste en mantener el nivel de rendimiento en condiciones específicas. Dado que el rendimiento se puede considerar como un hecho del dominio del modelo, la característica de fiabilidad concierne a un aspecto de la calidad semántica, es decir, a una correspondencia entre el modelo y el dominio.

De la misma manera, la *característica de eficiencia* definida como la capacidad de entregar un rendimiento apropiado a los recursos utilizados en condiciones específicas, corresponde a un aspecto de la calidad semántica, ya que el rendimiento según la utilización de los recursos es un hecho del dominio de modelado de los productos de software.

La *característica de usabilidad* definida en la ISO 9126, corresponde a la capacidad de los productos de software para ser comprendidos, aprendidos, utilizados y agradables para un usuario en un dominio específico. Dado que la característica de usabilidad necesita de usuarios que interactúen con el producto, esta característica se corresponderá con uno o varios tipos de calidad asociados al conjunto que tiene la interpretación de la audiencia (ver Figura 1), es decir, a las interpretaciones que pueden realizar los usuarios de los modelos conceptuales. La característica de usabilidad corresponde a un aspecto de la calidad pragmática, ya que si el modelo es comprendido y utilizado por los usuarios, existirá una correspondencia entre la externalización del modelo y la interpretación de la audiencia, es decir, se demostrará la calidad pragmática. Además, esta característica corresponde a un aspecto de la calidad de lenguaje, ya que si el modelo es com-

prendido, aprendido y utilizado, se demostrará la facilidad de entender, usar y aprender el lenguaje de modelado utilizado para realizar el modelo conceptual por los usuarios.

La *característica de mantenibilidad* consiste en poder realizar modificaciones a los productos de software. Estas modificaciones son necesarias para que los productos reflejen de mejor manera un dominio específico. Esta característica se puede entender como una propiedad del dominio de los modelos conceptuales, por lo que los modelos que cumplan con esta característica estarán contribuyendo a la calidad semántica.

La *característica de portabilidad* definida en la ISO 9126, consiste en la capacidad de los productos de software de ser transferidos de un entorno a otro, ya sea organizacional, de software o de hardware. Esta característica subsume la persistencia de los modelos de los productos de software, ya que es necesaria para poder realizar las transferencias a otros entornos. Por este motivo, la característica de portabilidad se corresponde con la perspectiva de calidad física, que plantea que los modelos sean persistentes y estén disponibles para el uso de la audiencia.

Dado que en la ISO 9126 no se especifican características de calidad asociadas a los lenguajes de modelado de los productos de software, como por ejemplo UML, la perspectiva de calidad sintáctica no queda correspondida con alguna de las características o sub-características del estándar ISO 9126.

Asimismo, el estándar ISO 9126 no define características de calidad asociadas al conocimiento de las personas que interactúan con los productos de software, por lo que no existen correspondencias directas entre las características definidas en el estándar con las perspectivas de calidad de conocimiento, semántica percibida y social definidas por Krogstie, ya que estos tipos de calidad están enfocados precisamente en el conocimiento de la audiencia, en las interpretaciones que realice la audiencia median-

te ese conocimiento y en el consenso de las interpretaciones realizadas por la audiencia.

Cabe destacar que, las características de calidad definidas en el estándar ISO 9126 sólo se reflejan en los tipos de calidad física, semántica, pragmática y de lenguaje, no cubriendo completamente los aspectos de estos tipos de calidad y dejando de lado los tipos de calidad semántica percibida, sintáctica, de conocimiento y social.

La Tabla 12 resume las relaciones existentes entre las perspectivas de calidad definidas por Krogstie y las características de calidad definidas en el estándar ISO 9126.

Tabla 12. Correspondencia entre perspectivas y características de calidad

Perspectiva de calidad Krogstie	Característica de calidad ISO 9126
Calidad Física	Portabilidad
Calidad Semántica	Funcionalidad Fiabilidad Eficiencia Mantenibilidad
Calidad Semántica Percibida	
Calidad Sintáctica	
Calidad del Lenguaje	Usabilidad
Calidad Pragmática	Usabilidad
Calidad del Conocimiento	
Calidad Social	

4.2 Análisis de métricas

El conjunto de propuestas de métricas presentado en la sección 3 contribuyen a la característica de *mantenibilidad*, dejando de lado las demás características definidas en el estándar ISO 9126. En base a esto, y al análisis de las características de calidad definidas en el estándar ISO 9126 presentado en la sección 4.1, se puede afirmar que las propuestas presentadas contribuyen a medir parcialmente aspectos de la perspectiva de *calidad semántica*, siendo no tratadas las otras perspectivas de calidad.

Se debe hacer hincapié en que las métricas presentadas sólo tratan la calidad semántica

de manera parcial, ya que para evaluar esta perspectiva de calidad, también se necesitan evaluar otras características de calidad, como son la funcionalidad, la fiabilidad y la eficiencia. Asimismo, es necesario destacar que las métricas presentadas no están relacionadas con todas las sub-características de la mantenibilidad. Como se puede ver en la Tabla 13, todas las propuestas están relacionadas con la sub-característica de *analizabilidad* y ninguna con la sub-característica de *facilidad de pruebas*.

La *analizabilidad* de los modelos conceptuales es la capacidad que tienen los productos de software de diagnosticar deficiencias en las partes identificadas para ser modificadas. Para el caso de los modelos conceptuales, la identificación de cada primitiva contribuirá a diagnosticar las deficiencias que pueden ocurrir cuando se modifiquen. Por este motivo, esta sub-característica está relacionada a todas las métricas presentadas en la sección 3. Por ejemplo, en la Tabla 13 se puede ver que seis de las seis métricas MOOD están relacionadas con esta sub-característica.

En menor medida que la sub-característica de *analizabilidad*, todas las propuestas presentan métricas para medir la *cambiabilidad* de los modelos conceptuales, es decir, la capacidad del modelo para permitir que una modificación sea realizada. Por ejemplo, en la Tabla 13 se puede ver que sólo una de las cuatro métricas de Li y Henry está relacionada a esta sub-característica. Del conjunto de propuestas presentadas, sólo las que presentan métricas para aplicarlas sobre modelos de clases contribuyen a la sub-característica de *estabilidad* de los modelos conceptuales, es decir, la capacidad de minimizar los efectos cuando se produce un cambio en el modelo. Por ejemplo, en la Tabla 13 se puede observar que cuatro de las doce métricas de Marchesi están relacionadas a esta sub-característica.

En cuanto a la sub-característica de *facilidad de pruebas*, tiene sentido que ninguna

de las propuestas presente métricas que estén relacionadas a esta sub-característica, debido a que dentro de los modelos no se tienen primitivas que evalúen los cambios que pueden realizarse en el dicho modelo.

Tabla 13. Proporción de contribución a sub-característica por propuesta

Propuestas	Anali-za-bilidad	Cam-bia-bilidad	Esta-bilidad	Facili-dad de pruebas
CK	6/6	4/6	3/6	0/6
Li y Henry	4/4	1/4	1/4	0/4
MOOD	6/6	2/6	1/6	0/6
Lorenz y Kidd	12/12	3/12	5/12	0/12
Marchesi	12/12	6/12	4/12	0/12
Harrison et al	1/1	1/1	1/1	0/1
Bansiya & Davis	11/11	7/11	5/11	0/11
Genero et al	16/16	8/16	6/16	0/16
JA Cruz-Lemus et al	9/9	1/9	0/9	0/9
Kiewkanya et al	6/6	2/6	0/6	0/6

Dado que las métricas relacionadas al estándar ISO 9126 sólo contribuyen a medir la calidad semántica de manera parcial, en la siguiente sección se describen cinco razones por las que estas métricas no son suficientes para asegurar la calidad de los modelos conceptuales desde una perspectiva industrial.

5. ASEGURAMIENTO DE LA CALIDAD EN MODELOS CONCEPTUALES: ALGUNAS RAZONES TÉCNICAS

El aseguramiento de la calidad de los modelos conceptuales depende de muchos factores, como por ejemplo factores técnicos, de proceso, sociales [1], etc. Por cada uno de los factores que contribuyen al aseguramiento de la calidad se pueden identificar razones por las que no se puede lograr este objetivo, sin embargo, en este artículo se muestran las razones técnicas:

Razón 1: Las métricas revisadas no contribuyen a la medición de la funcionalidad de los modelos conceptuales.

Desde una visión industrial, una de las características de calidad más importante es la funcionalidad, es decir, que el producto haga lo que tiene que hacer en un dominio determinado. Esta característica es importante debido a que si el producto hace lo que tiene que hacer, los usuarios generan confianza en la organización que ha desarrollado el producto, y probablemente comprarán nuevos productos a esa organización.

Para medir la característica de funcionalidad es necesaria la incorporación de modelos de requisitos, y mediante la trazabilidad entre los modelos de requisitos y los modelos conceptuales, identificar si se cumple con las funciones que debe realizar el software.

Otra forma de contribuir a la medición de la característica de funcionalidad es mediante el tamaño funcional de los modelos de requisitos y la comparación con el tamaño funcional de los modelos conceptuales.

Razón 2: Las métricas revisadas no contribuyen a la medición de la característica de usabilidad de los modelos conceptuales.

Otra característica importante para las organizaciones es la característica de usabilidad, ya que los usuarios deben entender como operar el producto de software para poder utilizarlo. En caso de que los productos sean difíciles de utilizar por los usuarios, aunque el producto presente todas las funcionalidades requeridas por ellos, el producto no será exitoso. De esta manera, los usuarios probablemente desearán el producto y no volverán a comprar productos a la misma compañía.

Para medir la característica de usabilidad en los modelos conceptuales, es necesario analizar los modelos de presentación, identificando las primitivas de los modelos de presentación (navegación, entidad, contexto, color, etc.) disponibles en cada ventana o página web, según sea la naturaleza del producto de software. Una vez identificadas

las primitivas, se pueden plantear métricas que permitan contribuir al aseguramiento de la característica de usabilidad.

Razón 3: Hoy en día, algunas características no pueden ser medidas en los modelos conceptuales, sino que deben ser medidas en los productos finales.

La característica de fiabilidad corresponde a que el producto mantenga un nivel específico de rendimiento, pero los modelos conceptuales no tienen rendimiento, por este motivo no se puede aplicar ni medir esta característica a los modelos conceptuales, sino que las mediciones deben realizarse en los productos finales.

La característica de eficiencia corresponde a que el producto tenga un nivel de rendimiento apropiado a los recursos utilizados. Como se dijo anteriormente, los modelos conceptuales no tienen rendimiento, por lo que esta característica también debe medirse en los productos finales.

La característica de portabilidad corresponde a la capacidad de los productos de ser transferidos de un entorno a otro. Los modelos conceptuales no tienen características del entorno de ejecución del producto de software que se generará con ellos, por lo tanto, la característica de portabilidad tampoco puede ser aplicada o medida en los modelos conceptuales.

Si bien las características de fiabilidad, eficiencia y portabilidad son importantes para las organizaciones, actualmente no pueden asegurarse en las etapas tempranas del ciclo de desarrollo de software. Sin embargo, existen métricas ([4], [5], [6]) que contribuyen a asegurar estas características en los productos finales.

Razón 4: No existe un modelo de calidad que permita medir todas las características de calidad definidas en el estándar ISO 9126.

Aunque existen varias propuestas que contribuyen a la medición de las características definidas en la ISO 9126, no existe un modelo de calidad que simplifique la unión

de todas esas propuestas, elimine las redundancias y agregue las métricas necesarias para medir estas seis características definidas, tanto a nivel de modelos conceptuales como de productos finales.

Razón 5: El estándar ISO 9126 no profundiza en las características sociales de las organizaciones.

A pesar de que se tuviera un modelo de calidad que permita medir todas las características definidas en el estándar ISO 9126, no se podría asegurar la calidad de los productos desde una visión industrial. Esto sucede porque las características de calidad definidas en el estándar ISO 9126 sólo se corresponden con algunas perspectivas de calidad, como son la calidad física, semántica, de lenguaje y pragmática, dejando de lado las perspectivas que reflejan la característica social de las organizaciones.

Por otro lado, algunas de las perspectivas de calidad no pueden ser medidas de forma cuantitativa, por lo que es necesario definir cuestionarios que permitan medirlas de forma cualitativa, según las percepciones de las personas que interactúen con los productos de software.

Por estos motivos, es necesaria la creación de un nuevo modelo de calidad, que permita evaluar todas las perspectivas de calidad para contribuir al aseguramiento de la calidad teniendo en cuenta la naturaleza social de las organizaciones.

6. CONCLUSIONES Y FUTURO TRABAJO

En este artículo se ha presentado dos modelos de calidad: el modelo de calidad de Krogstie y el estándar ISO 9126. Estos modelos conceptualizan la calidad de los productos de software de maneras diferentes. El modelo de calidad de Krogstie conceptualiza la calidad mediante la conjunción de los tipos de calidad que ha definido, teniendo en cuenta el constante cambio y la naturaleza social que tienen las organizaciones. Por otro

lado, el estándar ISO 9126 conceptualiza la calidad de los productos de software mediante la presencia de seis características de calidad, pero no tiene en cuenta la característica social de las organizaciones que utilizarán los productos.

Además, se ha presentado un conjunto representativo de métricas que permiten evaluar la calidad en etapas tempranas del ciclo de desarrollo de software. Para cada una de las métricas presentadas se han identificado las sub-características de calidad, y por consiguiente la característica de calidad, que tiene relacionadas. Se identificó que todas las propuestas ayudan a medir la característica de mantenibilidad, dejando de lado las demás características definidas en el estándar ISO 9126.

El análisis de las características definidas en la ISO 9126 según las perspectivas de Krogstie revela que el estándar no cubre todas las perspectivas que aseguran la calidad de los productos de software utilizados por organizaciones, debido a que sólo cubre parcialmente las perspectivas de calidad física, semántica, pragmática y de lenguaje, dejando de lado las perspectivas de calidad semántica percibida, sintáctica, de conocimiento y social.

Asimismo, el análisis de las métricas según las perspectivas de Krogstie refleja que sólo se contribuye de manera parcial a la calidad semántica, ya que la mantenibilidad es sólo uno de los aspectos de ese tipo de calidad.

Finalmente, se han identificado cinco razones por las que los modelos cuantitativos que contribuyen a la evaluación de las características definidas en el estándar ISO 9126, no son suficientes para asegurar la calidad de los modelos conceptuales desde una visión industrial. La principal razón es que el estándar ISO 9126 no toma en cuenta la característica social de las organizaciones. Por esta razón, es necesaria la creación de un modelo cuantitativo y cualitativo de calidad que permita asegurar la calidad de los mode-

los conceptuales y los productos finales desde todas las perspectivas de la calidad.

El trabajo futuro inmediato estará dado por la definición de un modelo cuantitativo de calidad, que permita asegurar la calidad de los productos de software según el estándar ISO 9126.

AGRADECIMIENTOS

Este trabajo ha sido desarrollado con el soporte del MEC bajo el proyecto DESTINO TIN2004-03534 y cofinanciado por FEDER.

REFERENCIAS

- [1] A. Gopal, T. Mukhopadhyay y M. S. Krishnan, "The Impact of Institutional Forces on Software Metrics Programs", *IEEE Transactions on Software Engineering* vol. 31, no. 8, pp. 679-694, 2005.
- [2] F. Brito e Abreu y W. Melo, "Evaluating the Impact of Object-Oriented Design on Software Quality", *3rd International Metric Symposium*, pp. 90-99, 1996.
- [3] ISO, "ISO/IEC 9126-1 – Software engineering – Product quality – Part 1: Quality Model", 2001.
- [4] ISO, "ISO/IEC 9126-2 – Software engineering – Product quality – Part 2: External Metrics", 2003.
- [5] ISO, "ISO/IEC 9126-3 – Software engineering – Product quality – Part 3: Internal Metrics", 2003.
- [6] ISO, "ISO/IEC 9126-4 – Software engineering – Product quality – Part 4: Quality in Use Metrics", 2004.
- [7] J.A. Cruz-Lemus, M. Genero y M. Piattini, "Metrics for Software Conceptual Models", Capítulo 7: Metrics for UML Statechart Diagrams, Imperial College Press, United Kingdom, 2005.
- [8] J. Bansiya y C. G. Davis, "A Hierarchical Model for Object-Oriented Design Quality Assessment", *IEEE Transactions on Software Engineering*, vol. 28, no. 1, pp. 4-17, 2002.
- [9] J. Krogstie, "Conceptual Modeling for Computerized Information Systems Support in Organizations", Tesis doctoral, Universidad de Trondheim, Trondheim, Noruega, 1995.
- [10] M. Genero, "Defining and Validating Metrics for Conceptual Models", Tesis doctoral, Universidad de Castilla-La Mancha, Madrid, España, 2002.
- [11] M. Genero, JA Cruz-Lemus y M. Piattini, "Construcción de un modelo de predicción para el entendimiento de los diagramas de estados en UML", Grupo ALARCOS, Departamento de Informática de la Uni-

versidad de Castilla- La Mancha, Ciudad Real, España, 2002.

[12] M. Genero, M. Piattini y C. Calero, "Empirical Validation of Class Diagram Metrics", International Symposium on Empirical Software Engineering, ISESE, 2002.

[13] M. Kiewhanya, N. Jindasawat y P. Muenchaisri, "A Methodology for Constructing Maintainability Model of Object-Oriented Design", 4th International Conference on Quality Software (QSIC'04), 2004.

[14] M. Lorenz y J. Kidd, "Object-Oriented Software Metrics: A Practical Guide", Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[15] M. Marchesi, "OOA Metrics for the Unified Modeling Language", 2nd Euromicro Conference on Software Maintenance and Reengineering, pp.67-73, 1998.

[16] OMG: Web page of OMG with companies with products MDA, <http://www.omg.org/mda/committed-products.htm>

[17] R. Harrison, S. Councell y R. Nithi, "Coupling Metrics for Object-Oriented Design", 5th International Software Metrics Symposium, pp. 150-156, 1998.

[18] S. Chidamber y C. Kemerer, "Towards a Metrics Suite for Object Oriented Design", Conference on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA'91), New York, USA, 1991.

[19] S. Chidamber y C. Kemerer, "A Metrics Suite for Object Oriented Design", IEEE Transactions on Software Engineering vol. 20, no. 6, pp. 476-493, 1994.

[20] Sitio web MDA, <http://www.omg.org/mda/>

[21] W. Li y S. Henry, "Maintenance Metrics for the Object Oriented Paradigm", 1st International Software Metrics Symposium, pp. 52-60, 1993.