

Atributos de Calidad para Componentes COTS: Una valoración de la información ofrecida por los vendedores.

Manuel F. Bertoa, José M. Troya y Antonio Vallecillo
Dpto. Lenguajes y Ciencias de la Computación. Universidad de Málaga.
{bertoa,troya,av}@lcc.uma.es

Resumen

La necesidad de desarrollar aplicaciones software complejas en periodos de tiempo cada vez más cortos está produciendo importantes cambios en la forma de construirlas, siendo el desarrollo de software basado en componentes (DSBC) uno de los mecanismos más efectivos para conseguir este objetivo. Esta nueva disciplina se basa en componentes software comerciales, construidos y probados previamente, que se conocen como componentes COTS (*Commercial-Off-The-Shelf*). Esto permite construir una aplicación buscando y ensamblando componentes, desarrollados por terceros y cuyo código no se puede modificar, que combinados adecuadamente satisfacen los requisitos del sistema.

Trabajos recientes de calidad software proponen un modelo de calidad específico para componentes, señalando las características y atributos de calidad más relevantes para este tipo de producto software. Para cada atributo existen diversas métricas que lo pueden cuantificar y que permiten describir y medir no sólo su funcionalidad sino también sus aspectos extrafuncionales. Por tanto, un punto importante del DSBC debe ser la documentación de los componentes. Sin embargo, la información que necesitan los desarrolladores para poder valorar y seleccionar un componente no suele estar disponible ni en los repositorios software ni en los sitios de los vendedores de componentes COTS.

En este trabajo valoramos y analizamos la información que los principales vendedores de componentes COTS ofrecen de sus productos, tratando de adecuar esta información con las métricas definidas para un modelo de calidad de componentes software. Nuestro objetivo ha sido estimar la diferencia existente entre la información que se necesita para realizar la selección y valoración de componentes y la información que ofrecen los principales vendedores de componentes software. Además, se ofrecen algunas recomendaciones y sugerencias para tratar de disminuir estas diferencias.

1 Introducción

El Desarrollo de Software Basado en Componentes (DSBC) aparece como una nueva disciplina que ayuda a los desarrolladores de software a realizar sus productos con menores esfuerzos tanto humanos como económicos. Su principal característica es utilizar componentes software ya desarrollados, que son combinados adecuadamente para satisfacer los requisitos del producto. Construir una aplicación se convierte por tanto en la búsqueda y ensamblaje de piezas prefabricadas, desarrolladas en su mayoría por terceras casas, y cuyo código no puede modificarse. Bajo este nuevo planteamiento, cobran especial interés los procesos de búsqueda y selección de los componentes apropiados para construir las aplicaciones.

Los principales esfuerzos de la comunidad de software en estos temas se han basado hasta ahora en los aspectos funcionales de los componentes, es decir, en la

funcionalidad que ofrecen. Sin embargo, por lo general se han venido obviando muchos de los aspectos de calidad que intervienen a la hora de seleccionar componentes. Este tipo de aspectos, que llamaremos “extra-funcionales”, cada vez acapara más la atención de los arquitectos e ingenieros del software. Por un lado, los requisitos extrafuncionales –por su naturaleza global– pueden afectar a varias partes del sistema, y por ello tendrán prioridad si entran en conflicto con los requisitos funcionales.

Añadido a todo esto, podemos encontrar una ausencia casi total de métricas que permitan dar una estimación más objetiva de estos atributos. Este hecho se ve afectado por la situación actual de los estándares internacionales relativos a la calidad del producto software. Las normas ISO 9126 [ISO, 2001] e ISO 14598 [ISO, 1999], encargadas de especificar estos temas, se encuentran ahora mismo en proceso de revisión. El proyecto SQuaRE [Azuma, 2001] es el encargado de tratar de hacerlas converger, eliminando algunas de las lagunas e inconsistencias que presentan. Por otro lado, es importante señalar que los estándares internacionales proporcionan guías y modelos de calidad para temas muy generales y su aplicación suele centrarse en temas de un gran espectro. Por tanto, no suelen estar pensadas para ofrecer soluciones en temas muy concretos, en particular para los relativos al DSBC y componentes COTS.

La siguiente causa de dificultad es la falta de información que suministran los vendedores de componentes software acerca de este tipo de atributos. Una visita virtual a los portales de los principales vendedores lo pone en evidencia, como ocurre con Componentsource (www.componentsource.com), Flashline (www.flashline.com) o WorldComp (www.wrlldcomp.com). Precisamente, el objetivo de este artículo se centra en obtener una visión general de la información que ofrecen los principales vendedores de componentes COTS y las posibilidades que ofrece esta información para ser utilizada en la evaluación de los atributos de calidad de un componente software dentro de un modelo de calidad específico para componentes.

El documento está estructurado en seis secciones. Tras esta introducción, la sección 2 hace una breve descripción del Desarrollo de Software Basado en Componentes. Después, la sección 3 propone un modelo de calidad para componentes software referenciando sus atributos de calidad. En la sección 4, se presentan los datos obtenidos en el análisis de una muestra de componentes ofertados por los principales vendedores de COTS. Por último, la sección 5 analiza los resultados encontrados y plantea algunas recomendaciones y trabajos futuros.

2 Desarrollo de Software Basado en Componentes (DSBC)

Antes de empezar, necesitamos definir qué es un componente software. En este artículo vamos a adoptar la definición que da Szyperski, donde un componente es “una unidad de composición de aplicaciones software que posee un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes, de forma independiente en tiempo y espacio” [Szyperski, 1998].

Una de las principales ventajas del desarrollo de software basado en componentes se basa en la “reutilización” de los mismos. De esta forma, los componentes se diseñan y desarrollan con el objetivo de poder ser reutilizados en otras aplicaciones, reduciendo el tiempo de desarrollo, mejorando la fiabilidad del producto final (al usar componentes ya probados previamente), y siendo más competitivos en costes.

Aunque hasta ahora la reutilización suele suceder principalmente en el ámbito interno de las organizaciones, el uso de los componentes comerciales comienza a extenderse. De esta forma se habla de componentes COTS, que han sido definidos como una clase especial de componentes software, normalmente de grano grueso, que presentan las siguientes características [Bass et al., 1999]:

- Son vendidos o licenciados al público en general
- Los mantiene y actualiza el propio vendedor, quien conserva los derechos de la propiedad intelectual
- Están disponibles en forma de múltiples copias, todas idénticas entre sí
- Su código no puede ser modificado por el usuario

La disponibilidad y uso cada vez mayor de este tipo de componentes está impulsando notablemente la creación de un mercado global de componentes COTS, que está pasando de ser la utopía de hace unos años a una realidad cada vez más cercana. La tecnología básica de componentes comienza a estar lo suficientemente madura (a través de plataformas de objetos distribuidos como EJB, CORBA o .NET) como para que numerosas empresas la adopten en sus nuevos desarrollos y sistemas de información. Incluso el gobierno y el ejército norteamericano han anunciado su uso, y han empezado a apostar por la utilización de componentes comerciales como única vía de mantener sus costes de desarrollo y mantenimiento de software bajo control [Sweeney et al., 2001]. Asimismo, están empezando a proliferar las empresas que venden con éxito componentes software al mercado general, como pueden ser *componentsource*, *flashline*, *worldcomp*, etc. [Seppanen y Helander, 2001].

3 Modelo de Calidad de Componentes

En general, no existe un consenso a la hora de definir y clasificar las características de calidad que debe presentar un producto software. Por tanto, hemos basado nuestra propuesta en los estándares internacionales, fundamentalmente el ISO 9126.

Siguiendo su terminología, entendemos por *característica* de calidad de un producto software a un conjunto de propiedades mediante las cuales se evalúa y describe su calidad. Una característica se puede refinar en múltiples niveles de subcaracterísticas.

Llamaremos *atributo* a una propiedad de calidad a la que puede asignársele una métrica, donde una métrica es un procedimiento que examina un componente y produce un dato simple, un símbolo (p.e. Excelente, Sí, No) o un número. Hay que tener en cuenta que no todas las propiedades que son mensurables (p.e. la “demostrabilidad”).

Un *modelo de calidad* es el conjunto de características y subcaracterísticas, y de cómo estas se relacionan entre sí. Por supuesto, el modelo de calidad a utilizar va a depender del tipo de producto a evaluar, en nuestro caso los componentes software.

Nuestro principal objetivo es detectar los atributos que pueden describir los proveedores (externos o internos) de componentes COTS en la información que suministran acerca de los mismos y que, por tanto, permitirían facilitar su valoración y selección por parte de los diseñadores y desarrolladores de productos software.

Partiendo del modelo de calidad general de ISO, vamos a tratar de particularizarlo realizando distintos tipos de clasificaciones.

1. En primer lugar, necesitamos discriminar entre aquellas características que tienen sentido para los componentes aislados (características locales) o bien deben ser valoradas a nivel de la arquitectura software de la aplicación (que llamaremos características globales). Por ejemplo, la “tolerancia a fallos” es una típica característica que va a depender de la arquitectura de la aplicación, mientras que la “madurez” es más propia de los componentes.
2. El instante en el cual una característica puede ser observada o medida, permite establecer otra clasificación de las características de un producto. Así, tenemos dos posibles categorías dependiendo de si la característica es observable en tiempo de ejecución (p.e. el rendimiento) o durante el ciclo de vida del producto (p.e. la mantenibilidad) [Preiss et al., 2000].
3. Como se menciona en los estándares de ISO, es importante identificar los usuarios a los que se dirige el modelo. En nuestra propuesta, los usuarios son fundamentalmente los arquitectos software, que necesitan evaluar los componentes COTS que van a formar parte de su aplicación. Así, las interfaces de los componentes objeto de nuestro estudio son más las interfaces programáticas (es decir, las APIs que definen las formas de acceder desde otros programas a los servicios que ofrecen los componentes), que las interfaces de usuario.
4. Para componentes COTS, es fundamental distinguir entre métricas internas y externas. Las internas miden los atributos internos del producto final o de los productos intermedios (p.e. la especificación o el código fuente) durante el diseño y la codificación. Las externas son las que realizan las mediciones en función del comportamiento del sistema durante las pruebas y la operación del componente. Por tanto, debido al carácter de caja negra de los componentes COTS, son las métricas externas las que interesan. Esto no quita que algunas de las características internas den una medida indirecta de las externas, e incluso que puedan tener efectos sobre la arquitectura final. Así por ejemplo, el tamaño de un componente puede ser importante a la hora de tener en cuenta los requisitos de espacio de la aplicación.

Por último, es importante reseñar que además de las características de calidad en un componente, hay otro conjunto de características no relacionadas directamente con la calidad como pueden ser el precio, la asistencia técnica, las condiciones de licencia, etc., que también son necesarias a la hora de valorar el componente más adecuado, pero que quedan fuera del ámbito de este artículo.

Como hemos indicado anteriormente, no todas las características de un producto software son aplicables a un componente COTS. La Tabla 1 muestra el modelo de calidad que proponemos para este tipo de producto software. Básicamente se trata del modelo de calidad de ISO, donde desaparecen algunas subcaracterísticas de Mantenibilidad y Portabilidad y las subcaracterísticas Tolerancia a Fallos, Estabilidad y Analizabilidad. Es importante observar que algunas de ellas cambian de sentido, como detallamos a continuación (las características que cambian su sentido aparecen en **negrita** en la Tabla 1).

Característica	Subcaracterística (tiempo ejecución)	Subcaracterística (ciclo de vida)
Funcionalidad	Precisión Seguridad	Idoneidad Interoperatividad Conformidad
Fiabilidad	Recuperabilidad	Madurez
Facilidad de Uso		Facilidad de Aprendizaje Facilidad de Comprensión Operatividad
Eficiencia	Comportamiento Temporal Utilización de Recursos	
Mantenibilidad		Cambiabilidad Facilidad de Prueba
Portabilidad		Reemplazabilidad

Tabla 1. Modelo de Calidad para COTS

Funcionalidad. Esta característica mantiene el mismo sentido para los componentes que para un producto software. Podríamos expresarla como la capacidad del componente para proporcionar las funciones que satisfagan las necesidades establecidas o implícitas cuando se usa bajo las condiciones especificadas.

Fiabilidad. Es aplicable directamente a los componentes, y fundamental para su reutilización. La subcaracterística de Madurez la medimos en función de los cambios que sufren las versiones comerciales y la velocidad a la que aparecen. La Recuperabilidad, en función de una serie de atributos que pueden estar presentes o no en su diseño, indicando los métodos que se utilizan para implementarlos.

Facilidad de Uso. Esta característica, y todas sus subcaracterísticas cambian de sentido, dado que un componente no será utilizado por un usuario final directamente sino por los diseñadores y desarrolladores de aplicaciones software. La facilidad de uso real de un componente debe interpretarse como la capacidad del componente para ser utilizado en la construcción de un producto o sistema software. En este sentido buscaremos atributos que midan la facilidad de uso del componente durante el diseño de las aplicaciones.

Eficiencia. Vamos a respetar la definición y clasificación que hace la ISO de esta característica (en Comportamiento Temporal y Utilización de Recursos), aunque la mayoría de las propuestas de otros autores prefieren hablar de Rendimiento (*Performance*) y usan otra subclasificación. En cualquier caso, los atributos que vamos a definir para medir estas características son aplicables de forma independiente al nombre y clasificación que se utilice.

Mantenibilidad. La facilidad de mantenimiento o *mantenibilidad* mide la capacidad de un producto software de ser modificado, entendiéndose por modificación cualquier corrección, mejora o adaptación del software. En este sentido, y aunque las

modificaciones internas no serán realizadas por el usuario del componente (desarrollador), sí necesitará probar el componente antes de incluirlo en su aplicación o cambiar alguno de los parámetros que se pueden particularizar. Por ello, las subcaracterísticas Cambiabilidad y Facilidad de Prueba son las que deben ser medidas para los componentes.

Portabilidad. Esta característica se define como la capacidad del producto software para poder ser reutilizado en distintos entornos. En COTS, esa es la esencia misma de los componentes, que son diseñados y desarrollados específicamente para ser reutilizados. (Es importante observar que en la Ingeniería del software Basada en Componentes, reutilizar significa no sólo usar más de una vez, sino usar en distintos contextos [Szyperski, 1998]).

Las tres primeras columnas de la Tabla 4 muestran los atributos de calidad para componentes COTS mensurables en tiempo de ejecución. Similarmente, las tres primeras columnas de la Tabla 5 muestran los atributos de calidad mensurables durante todo el ciclo de vida. Una descripción detallada del modelo de calidad para componentes y de los atributos de calidad se puede encontrar en [Bertoa y Vallecillo, 2002].

4 Análisis de la Información Ofrecida por lo Vendedores de COTS

Para tener una visión global de la información que ofrecen los vendedores de COTS hemos analizado una muestra de 96 componentes de varios vendedores de COTS, siendo la muestra más significativa (64) la de *componentsource*. El método de elección de los componentes de la muestra consistió en seleccionar la mitad de las categorías que aparecen en el catalogo del vendedor y escoger para cada una de ellas dos componentes. Uno, el más vendido, indicado en una lista de los 5 más vendidos dentro de cada categoría, y el segundo de forma aleatoria entre los componentes de cada categoría. Si este coincidía con el anterior o era del mismo fabricante, seleccionamos el siguiente en la lista con el objetivo de abarcar un numero mayor de fabricantes.

La primera dificultad detectada es que la información necesaria para evaluar los atributos de calidad no está disponible de forma fácil. Es necesario buscarla entre los ficheros de ayuda, los manuales o las demostraciones que ofrecen los fabricantes. Por tanto, cualquier intento de realizar una evaluación más o menos automatizada es prácticamente imposible.

No obstante, en la página de cada producto aparece actualmente un conjunto de datos, denominados datos valorables (*asset values*) como Líneas de Código, Tiempo dedicado a I+D o un factor de madurez de la empresa (*man months skill factor*). El grado de cumplimentación de estos tres datos que ofrece *componentsource* se resume en la Tabla 2 para los componentes analizados.

N=64	Man Months R&D	Man Months Skill Factor	Lines of Code
SI	22%	20%	20%
NO	78%	80%	80%

Tabla 2. Porcentaje de fabricantes COTS que dan información sobre “datos valorables”

El siguiente conjunto de información que hemos analizado es el tipo de información para la evaluación de los productos que ofrecen los desarrolladores de COTS. En la muestra analizada aparecen tres tipos de ficheros con información relevante: Ficheros de Ayuda y/o Manuales; Productos de evaluación; y Diagramas UML.

Los Ficheros de Ayuda ofrecidos van desde algunas páginas de publicidad del producto hasta el manual completo o guías de usuario. Hemos considerado este tipo de ficheros como adecuados cuando la información que aportan puede utilizarse para evaluar alguno de los atributos del modelo de calidad.

Los productos de evaluación (o demos) se presentan también con diversos formatos, proporcionando la funcionalidad completa durante un corto periodo de tiempo (p.e.: 30 días) o limitando algunas funcionalidades o capacidades. En cualquier caso, aunque la información que ofrece una demo, desde el punto de vista de una evaluación inicial de los atributos de calidad, no es muy adecuada ya que obliga a la instalación del producto y a un análisis laborioso de sus funcionalidades y atributos de calidad, sí nos parece importante resaltar la posibilidad de probar el producto.

Por último, los diagramas UML se muestran como un fichero independiente o de forma menos detallada dentro del manual o del fichero de ayuda, por ejemplo, proporcionando sólo un diagrama de objetos. En cualquier caso, dado que es una información poca habitual pero quizás la más interesante para evaluar los componentes, hemos recogido este tipo de información cuando aparece algo.

La Tabla 3 muestra el porcentaje de componentes COTS que tenían disponibles estos tipos de información según los datos obtenidos en nuestro análisis:

N=96	Ficheros Ayuda	Demo	Diagramas UML
SI	86%	81%	10%
NO	14%	19%	90%

Tabla 3: Información para evaluación de COTS

La siguiente información analizada ha sido si es posible evaluar de alguna forma los atributos de calidad que proponemos en el modelo de calidad para componentes software tomando como base la información ofrecida en la página de cada componente mostrada por el vendedor o recogiénola de los ficheros de ayuda, manuales y demostraciones que se pueden descargar desde dicha página.

Hemos analizado las posibilidades de que la información ofrecida nos permita realizar alguna métrica para los atributos propuestos, y en este sentido hemos evaluado si la información era adecuada o no. Las siguientes tablas muestran los resultados obtenidos, donde hemos mantenido la clasificación de atributos mensurables en tiempo de ejecución (Tabla 4) y mensurables durante todo el ciclo de vida (Tabla 5).

En estas tablas, la columna “SI” indica el porcentaje de COTS donde la información facilitada por el fabricante permite valorar alguna métrica relacionada con el atributo. Un valor de 0% en dicha columna, indica que ninguno de los COTS analizados proporciona información para evaluar ese atributo. Las subcaracterísticas y atributos con un porcentaje del 50% o superior se resaltan en negrita.

Característica	Subcaracterística	Atributo	SI(%)
Funcionalidad	Precisión	Precisión	0
		Exactitud Computacional	0
	Seguridad	Cifrado de Datos	5
		Capacidad de Control	0
		Capacidad para Auditar	5
Fiabilidad	Recuperabilidad	Secuenciable	0
		Persistente	5
		Transaccional	10
		Tratamiento de Errores	40
Eficiencia	Comportamiento Temporal	Tiempo de Respuesta	0
		Capacidad de Emisión	0
		Capacidad de Recepción	0
	Utilización de Recursos	Requisitos de Memoria	81
		Utilización de Disco	95

Tabla 4 Información Disponible para Evaluar los Atributos de Calidad de Tiempo de Ejecución

5 Conclusiones y Recomendaciones

La mayoría de los atributos de tiempo de ejecución son muy difíciles de evaluar dado la ausencia casi total de información relativa a los mismos. La excepción es la subcaracterística de Utilización de Recursos debido a que las necesidades de disco y memoria aparecen directamente en la página del vendedor. Otro atributo con un porcentaje bajo (40%) pero significativo en este contexto es el Tratamiento de Errores dado que suele aparecer información relacionada en los manuales o guías. El resto de atributos que tienen un porcentaje de COTS pequeño pero superior a cero aparecen debido a que su propia funcionalidad esta relacionada con el atributo, por ejemplo, uno de los componentes analizados proporcionaba persistencia a otros componentes o aplicaciones.

La situación es diferente en los atributos de ciclo de vida, donde observamos dos grupos de subcaracterísticas. Un grupo donde la evaluación es muy difícil debido a la ausencia de información relacionada con sus atributos o con un porcentaje de COTS donde existe información muy bajo (20% o menor). En este grupo aparecen Conformidad Funcional, Madurez, Facilidad de Aprendizaje, Facilidad de Prueba y Reemplazabilidad. En algunas de estas características, como Facilidad de Aprendizaje, la falta de información se explica porque los atributos propuestos deben evaluarse por terceras casas o por expertos independientes. Igualmente ocurre en el caso de los estándares y de la certificación. Más llamativo es el caso de la evolucionabilidad y volatilidad de las versiones, que son unos datos fácilmente reseñables por los fabricantes. También es destacable la falta de elementos de prueba en la gran mayoría de COTS analizados.

El segundo grupo de subcaracterísticas esta formado por aquellas cuyos todos sus atributos o la mayoría de ellos presentan un porcentaje del 50% o superior de COTS con información relativa a los mismos. En este grupo podemos incluir las subcaracterísticas de Idoneidad, Interoperatividad, Facilidad de Comprensión, Operatividad y Cambiabilidad.

Las subcaracterísticas de Idoneidad, Operatividad y Cambiabilidad se pueden evaluar conociendo las interfaces, métodos, propiedades y eventos, este tipo de información se puede extraer cuando el fabricante de COTS proporciona el manual o

una ayuda detallada lo cual aparece aproximadamente en la mitad de los casos analizados. Esta forma de obtener la información necesaria para la evaluación no es obviamente la ideal pero al menos se puede obtener aunque con un trabajo laborioso.

Característica	Subcaracterística	Atributo	SI (%)
Funcionalidad	Idoneidad	Cobertura	55
		Exceso	55
		Cobertura de la Implementación	55
	Interoperatividad	Compatibilidad de los Datos	95
	Conformidad	Conformidad con Estándares	0
		Certificaciones	0
Fiabilidad	Madurez	Volatilidad	5
		Evolucionabilidad	10
		Fallos eliminados	15
Facilidad de Uso	Facilidad de Aprendizaje	Periodo para usar correctamente	0
		Periodo para configurar correctamente	0
		Periodo para administrar correctamente	0
		Periodo para dominar	0
	Facilidad de Comprensión	Documentación de Usuario	65
		Sistema de Ayuda	65
		Documentación Computacional	0
		Formación	30
	Operatividad	Cobertura de la Demostración	90
		Interfaces Ofrecidas	55
		Interfaces externas utilizadas	50
		Índice de Complejidad	50
		Esfuerzo para operar	0
		Esfuerzo para configurar	50
Mantenibilidad	Cambiabilidad	Esfuerzo para administrar	0
		Modificabilidad	55
		Índice de Modificabilidad	55
	Facilidad de Prueba	Capacidad de Control de Cambio	55
		Auto-Test de Arranque	0
Portabilidad	Reemplazabilidad	Batería de Pruebas	5
		Compatibilidad hacia atrás	20

Tabla 5 Información Disponible para Evaluar los Atributos de Calidad de Tiempo de Ejecución

Los datos sobre compatibilidad aparecen en la página inicial, lo cual da un alto porcentaje de COTS que suministran esta información.

La subcaracterística de Facilidad de Comprensión muestra que algunos de sus atributos se pueden evaluar y otros no. Entre los primeros están los relativos a las demostraciones (95%) y los de ayuda y documentación (65%), los que carecen de información relativa son los relativos a la documentación procesable automáticamente y a la formación.

A la vista de estos resultados, la información que los vendedores ofrecen directamente en la página inicial de cada producto debe recoger una mayor cantidad de datos que permita iniciar una evaluación de los atributos de calidad. En este sentido, parece de gran interés la inclusión de los datos de valorables (asset values) que ofrece *componentsource*. Aunque esto podría ser un primer paso, estaríamos en

una valoración y evaluación manual de los posibles candidatos. El objetivo debe ser que esta información pueda evaluarse de forma automática lo cual facilitaría la selección de componentes.

La necesidad de disponer de información para evaluar componentes no puede hacernos perder de vista la dificultad que tiene para los fabricantes y vendedores proporcionarla, por tanto, otra meta debe ser el consensuar métricas fáciles de medir y cuyos datos se pueden obtener sin necesidad de complicados métodos y operaciones. Si observamos que ningún fabricante de COTS suministra información relativa a un atributo de calidad, debemos reflexionar si dicho atributo es necesario o se puede reemplazar por otro que sea más fácil de valorar.

Por último, pensamos que para conseguir una verdadera ingeniería del software de componentes es necesario disponer de métricas que cuantifiquen sus atributos de calidad. El panorama descrito en este artículo no es muy alentador en este sentido pero para resolver este conflicto a largo plazo podemos basarnos en dos ideas. En primer lugar, es difícil conseguir un acuerdo sobre temas demasiado genéricos. Por ello, es preciso definir y consensuar modelos de calidad para productos específicos, en nuestro caso para componentes software. Y en segundo lugar, quizás la evaluación de los atributos de calidad de los componentes debería realizarse por entidades independientes, al menos hasta que los fabricantes de componentes software alcance la madurez que existe en la industria de componentes hardware donde los propios fabricantes ofrecen para sus productos catálogos (*data sheet*) con los valores de sus atributos de calidad. Hoy en día parece una utopía disponer de este tipo de catálogos para componentes software pero si realmente deseamos hacer una “ingeniería” del software basada en componentes que produzca productos de alta calidad ésta debe ser la tendencia.

Referencias

- [Azuma, 2001] M. Azuma “SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality”. In ESCOM (European Software Control and Metrics conference), April 2001.
- [Bass et al., 1999] L. Bass and P. Clements y R. Kazman. Software Architecture in Practice. 6ª edición. AddisonWesley, 1999.
- [Bertoa y Vallecillo, 2002] M. F. Bertoa y A. Vallecillo. "Quality Attributes for COTS Components". In Proc. of the 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2002). Málaga, Spain, June 2002. Disponible en: <http://www.lcc.uma.es/~av/Publicaciones/02/COTSmetrics.doc>
- [ISO, 1999] ISO/IEC 14598-1 “Information Technology – Software Product Evaluation – Part 1: General overview”. May, 1999.
- [ISO, 2001] ISO/IEC 9126-1:2001 “Software Engineering – Product Quality – Part 1: Quality model”. June, 2001.
- [Preiss et al., 2001] O. Preiss, A. Wegmann, y J. Wong. “On Quality Attribute Based Software Engineering”, In Proc. of the 27th Euromicro Conference, Warsaw, Poland, September 2001. IEEE CS Press.
- [Sedigh et al., 2001] S. SedighAli, A. Ghafoor y R.A. Paul. “Software Engineering Metrics for COTS based Systems”. IEEE Computer, 34(5):4450, May 2001.
- [Seppanen y Helander, 2001] V. Seppanen y N. Helander. “Original Software Component Manufacturing: Survey of the StateofthePractice”. 27th Euromicro Conference, pp. 138-145. Varsovia, Polonia, September 2001.
- [Sweeney et al., 2001] L.E. Sweeney, E.V. Kortright y R.J. Buckley. “Developing an RMODP based architecture for the defense integrated military human resources system”. J. Cordeiro and H. Kilov (eds.) Proceedings of WOODPECKER’01, pp. 110-124. Setúbal, Portugal, July 2001.
- [Szyperski, 1998] C. Szyperski. Component Software. Beyond Object Oriented Programming. AddisonWesley Longman, 1998.