



REICIS. Revista Española de Innovación,
Calidad e Ingeniería del Software

E-ISSN: 1885-4486

reicis@ati.es

Asociación de Técnicos de Informática
España

Marcos, José; Arroyo, Alicia; Garzás, Javier; Piattini, Mario
La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre
REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software, vol. 4, núm. 2,
septiembre, 2008, pp. 133-144
Asociación de Técnicos de Informática
Madrid, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=92218339013>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre

José Marcos¹, Alicia Arroyo¹, Javier Garzás^{1 y 2}, Mario Piattini³

¹Kybele Consulting SL

{Jose.Marcos, Javier.Garzas, Alicia.Arroyo} @kybeleconsulting.com

²Grupo de Investigación Kybele. Universidad Rey Juan Carlos

Javier.Garzas @urjc.es

³Grupo Alarcos. Dpto. de Tecnologías y Sistemas de Información.

Universidad de Castilla-La Mancha

Mario.Piattini@uclm.es

Abstract

The quality control should be desirable done from a quantitative point of view, for which it is necessary to establish measurement systems throughout the product lifecycle. Code metrics provide the basis for the improvement and, in particular the product, provide direct visibility of product quality. However, in order to do efficient the quality systems, they should have a high level of automation and provide information with varying degrees of detail to the different actors surrounding the product. With these purposes, this paper presents technological and methodological measurement infrastructure that can be executed in a specific way or in an environment of continuous integration and which is based on the standard ISO/IEC 9126, providing different levels of product quality information.

Key words: quality, measurement, quality control panels.

Resumen

El control de la calidad se debe realizar desde un punto de vista cuantitativo, para lo cual es necesario establecer sistemas de medición durante todo el ciclo de vida del producto. Las métricas de código proporcionan la base para la mejora; en concreto, las de producto proporcionan visibilidad directa de la calidad del producto. Sin embargo, para que los sistemas de calidad sean eficientes, deben tener un alto grado de automatización y proporcionar información con diferente grado de detalle a los diferentes actores que rodean al producto. Con este objetivo, este artículo presenta infraestructura tecnológica y metodológica de medición que se puede ejecutar de forma puntual o en un entorno de integración continua, y que se basa en el estándar ISO/IEC 9126, proporcionando de esta forma diferentes niveles de información para la calidad del producto de software.

Palabras clave: calidad, métricas, cuadros de mando de calidad.

1. Introducción

“Quality is a complex and multifaceted concept.” La calidad puede describirse desde diferentes perspectivas [1]; en el desarrollo del software hay dos especial y

tradicionalmente importantes: la calidad del producto en sí y la calidad del proceso para obtenerlo (o actividades, tareas, etc., para desarrollar y mantener software). Dos dimensiones esenciales, estudiadas desde hace tiempo [2] y que giran e interactúan en torno a la idea de que la calidad del producto está determinada por la calidad del proceso usado para desarrollarlo y mantenerlo.

No obstante, hoy en día parecen estar más de moda los modelos de mejora de procesos (CMMI, ISO 12207 e ISO 15504, principalmente), como reflejan recientes informes [3, 4] que, aunque incluyen actividades de medición de procesos y productos, se centran en los procesos. Sin embargo, como hace tiempo comentaban Kitchenham y Pfleeger[5], la principal crítica a esta visión es que hay poca evidencia en que cumplir un modelo de procesos asegure la calidad del producto; además, la estandarización de los procesos garantiza la uniformidad en la salida de estos, lo que puede incluso institucionalizar la creación de malos productos. Más recientemente, Maibaum y Wassynng [6], en esta línea, comentaban que las evaluaciones de calidad deberían estar basadas en evidencias extraídas directamente de los atributos del producto, no en evidencias circunstanciales deducidas desde el proceso.

En este artículo se muestra una implementación de la parte referente a la mantenibilidad de la norma ISO/IEC 25000 [7], haciendo uso de herramientas de software libre, lo que permite obtener una medida de la calidad del producto de software. El entorno, denominado KEMIS (Kybele Enviroment Masurement Information System), proporciona una infraestructura para dicha medición y que se puede ejecutar de forma puntual o integrada en entornos de integración continua, permitiendo obtener de forma automática y periódica un conjunto de informes relativos a la calidad del producto, obteniendo métricas de código y microarquitectura, dentro del marco de la norma 25000. Las restricciones de KEMIS vienen dadas por la parte de la mantenibilidad que se incluye en la división 2502n. Además, varias empresas importantes (empresas públicas, una administración regional y una administración pública, entre otras) están usando este entorno para la evaluación de productos de software.

En el epígrafe segundo se tratan las normas ISO encargadas de la calidad del producto de software; en el apartado tercero, las métricas de producto y su relación con las herramientas de software libre; en el cuarto, el proyecto KEMIS y su relación con la calidad

del producto; en el apartado quinto se presenta un ejemplo de evaluación y, por último, en el apartado sexto se ofrecen las conclusiones.

2. Las normas ISO para la calidad de un producto de software

En el año 1991 la ISO (International Organization for Standardization) publicó su modelo de calidad para la evaluación del producto de software (ISO 9126:1991), que fue extendiendo con revisiones hasta 2004, dando lugar a la actual norma ISO/IEC 9126 “Software Engineering. Product Quality” [8]. La norma ISO/IEC 9126 propone un conjunto de características, subcaracterísticas y atributos para descomponer la calidad de un producto de software. Propone seis propiedades (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), que se dividen en subcategorías, como se muestra en la Figura 1.

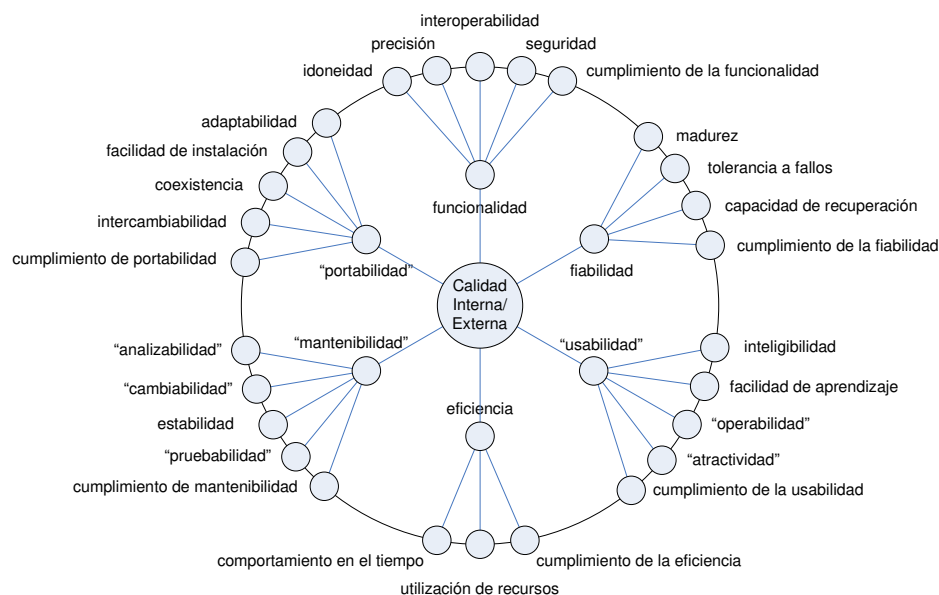


Figura 12. Características de la calidad interna y externa según la ISO/IEC 9126.

Recientemente ha aparecido una nueva versión de la norma 9126: la norma ISO/IEC 25000. Esta proporciona una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE). Constituyen una serie de normas basadas en la ISO 9126 y en la ISO 14598 [9] (Evaluación del Software), y su objetivo principal es guiar el desarrollo de los productos de

software con la especificación y evaluación de requisitos de calidad. Establece criterios para la especificación de requisitos de calidad de productos de software, sus métricas y su evaluación. Incluye un modelo de calidad dividido en dos partes para unificar las definiciones de calidad de los clientes con los atributos en el proceso de desarrollo. SQuaRE está formada por las divisiones siguientes:

- ISO/IEC 2500n. División de gestión de calidad. Los estándares que forman esta división definen todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.
- ISO/IEC 2501n. División del modelo de calidad. El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.
- ISO/IEC 2502n. División de mediciones de calidad. Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto de software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.
- ISO/IEC 2503n. División de requisitos de calidad. Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto de software que va a ser desarrollado ó como entrada para un proceso de evaluación. El proceso de definición de requisitos se guía por el establecido en la norma ISO/IEC 15288.
- ISO/IEC 2504n. División de evaluación de la calidad. Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto de software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.
- ISO/IEC 25050–25099. Estándares de extensión SQuaRE. Incluyen requisitos para la calidad de productos de software “*Off-The-Self*” y para el formato común de la industria (CIF) para informes de usabilidad.

La norma ISO 25000 ha sido desarrollada por el subcomité SC 7 (Ingeniería de software y sistemas) del Comité Técnico Conjunto ISO/IEC JTC 1.

3. Las métricas de producto y su medición con software libre

Los citados modelos de calidad del producto se basan en el concepto de métrica como base para su aplicación. En lo que se refiere a métricas, una de las más representativas es son las Líneas de Código (LOC, Lines of Code).[10] Otra métrica de especial relevancia es la Complejidad Ciclomática (CC, Cyclomatic Complexity), definida por McCabe[11], que se utiliza para evaluar la complejidad de un programa. Desde 1976 ha aparecido una gran cantidad de métricas, entre las que podemos destacar las definidas por Chidamber y Kemerer[12] o las que establecen Brito e Abreu y Carapuça[13], que se usan dentro del paradigma de Orientación a Objetos.

Un importante avance en este sentido es que en la actualidad existe un amplio abanico de herramientas de software libre que permiten obtener dichas métricas de producto. Entre ellas destacan JavaNCSS (www.kclee.de/clemens/java/javancss), PMD (<http://pmd.sourceforge.net>), Checkstyle (<http://checkstyle.sourceforge.net>), Findbugs (<http://findbugs.sourceforge.net>) y JDepend (www.clarkware.com) que, junto con herramientas de construcción de software como Maven (<http://maven.apache.org/>) y Ant (<http://ant.apache.org/>), y de integración continua como Cruise Control (<http://cruisecontrol.sourceforge.net/>) o Continuum (<http://continuum.apache.org/>), permiten automatizar la obtención de las métricas, personalizar los resultados y definir la periodicidad con la que se realizan las mediciones.

4. El proyecto KEMIS: un entorno de software libre para la medición de la mantenibilidad del producto

"En ocasiones la proporción de coste que supone el mantenimiento ronda el 90%"[14]. El mantenimiento en el ciclo de vida del software se considera el proceso con más peso económico, debido a que se basa en la labor de corrección de errores durante un largo período de tiempo y en la facilidad con la que se puede añadir funcionalidad. Cabe destacar que el mantenimiento del producto está estrechamente relacionado con la fiabilidad del sistema (el tiempo de corrección de un fallo determina el tiempo en que no se puede usar una funcionalidad o un sistema completo), que es algo que incide directamente sobre el usuario y el cliente.

La aportación principal de KEMIS es ofrecer indicadores de mantenibilidad que aporten visibilidad durante el desarrollo, la adquisición y el mantenimiento de un producto, proporcionando una infraestructura de medición automática, que se basa en métricas que se pueden obtener con herramientas de software libre y el modelo de calidad que propone la ISO/IEC 9126.

4.1. Atributos de calidad

El modelo de calidad de la ISO/IEC 9126 establece que las características están compuestas de subcaracterísticas y estas, a su vez, de atributos de calidad.

Los atributos cobran especial relevancia, ya que son el punto de unión entre el modelo de calidad y las métricas disponibles, en especial, las más maduras y ampliamente aceptadas, que hoy en día se pueden obtener con herramientas de software libre.

El modelo se basa, por tanto, en la definición de atributos de calidad que se puedan obtener directamente del producto. En KEMIS no solo se ha realizado una selección de atributos de calidad relativos a subcaracterísticas de la mantenibilidad, sino que se han establecido las funciones necesarias para obtenerlos a partir de métricas de código.

Característica	Subcaracterística	Atributo
Mantenibilidad	Capacidad para ser analizado	Densidad de complejidad ciclomática Densidad de código repetido Densidad de comentarios Densidad de defectos de capacidad de ser analizado
	Capacidad para ser cambiado	Densidad de defectos de facilidad de cambio Densidad de dependencias cíclicas
	Estabilidad	Densidad de defectos de estabilidad
	Capacidad para ser probado	Densidad de defectos en pruebas unitarias Densidad de pruebas unitarias Cobertura de pruebas unitarias Tasa de fallos de pruebas unitarias Tasa de errores fe pruebas unitarias
	Cumplimiento de la mantenibilidad	Densidad de defectos de mantenibilidad definidos por la empresa

Tabla 3. Relación de atributos contemplados en KEMIS.

En la Tabla 1 se recoge la selección de atributos de calidad realizada y las subcaracterísticas de calidad a las que afectan.

4.2. Normalización de atributos de calidad

Los atributos de calidad reflejan propiedades heterogéneas del producto de software; por eso, al manejarlos para componer una subcaracterística, es deseable que estén normalizados y que sus posibles valores pertenezcan a los mismos rangos con el mismo significado.

Una de las aportaciones de KEMIS es que proporciona funciones para normalizar los indicadores de atributos de calidad, de tal forma que se permite su escalado a subcaracterísticas y características de calidad, además de facilitarse su interpretación. En este sentido, antes de pasar a componer una subcaracterística, los atributos se pasan por una función de calidad cuyo objetivo es proporcionar un valor de calidad entre 0 y 100.

Densidad de complejidad ciclomática	
Nivel	Táctico
Modelo de análisis	de $DenCC = \frac{\sum CC}{NCSS} \times 100$
Atributos	DenCC: Densidad de la complejidad ciclomática CC: Complejidad ciclomática NCSS: Non Commenting Source Statements (sentencias de código fuente que no son comentario)
Función de nivel de calidad	$Calidad(DenCC) = \begin{cases} 0 & , si \quad DenCC \leq m \\ \frac{DenCC - m}{n - m} \times 100 & , si \quad m \leq DenCC \leq n \\ 100 & , si \quad n \leq DenCC \leq u \\ \frac{v - DenCC}{v - u} \times 100 & , si \quad u \leq DenCC \leq v \\ 0 & , si \quad DenCC > v \end{cases}$
Umbrales recomendados	$m = 0,08; n = 0,16; u = 0,24 \text{ y } v = 0,36$
Criterio de calidad	Los valores inferiores al umbral m o superiores al umbral v serán considerados como nulos. Los valores que sean mayores que el umbral n y menores que el umbral u serán considerados como óptimos. El resto de valores serán proporcionales a la distancia entre los valores nulo y óptimo.
Interpretación del indicador	Los valores cercanos a 0 indican un nivel deficiente de calidad en el atributo. Los valores cercanos a 100 indican un valor alto en él.

Tabla 4. Normalización de la densidad de complejidad ciclomática.

En la Tabla 4, a modo de ejemplo, se muestra una definición del atributo Densidad de complejidad ciclomática: la forma de calcularla, su función de calidad, los criterios que se han adoptado para conseguir esa función y los umbrales asociados.

La función de calidad resultante se puede ver en la Figura 2.

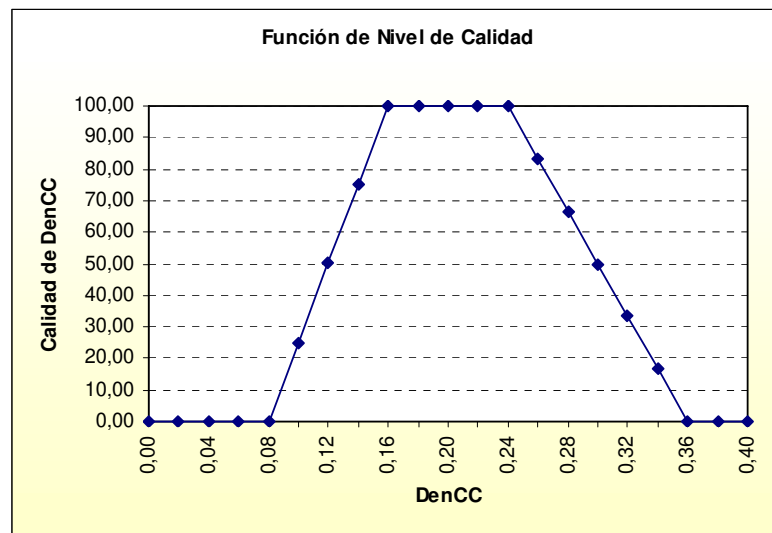


Figura 2. Función de calidad de la densidad de la complejidad ciclométrica.

Esta función permite normalizar atributos de calidad. Mientras que para la Densidad de la complejidad ciclométrica podemos obtener valores que oscilan en un rango entre 0 y 1, existen otros atributos que pueden tener otros rangos; por ejemplo, entre 0% y 100%, por lo que es necesaria una normalización de cada atributo.

4.3. Escalado de indicadores de calidad

En KEMIS, una vez definidos y normalizados los atributos, se hace uso de unas funciones que permiten componer las subcaracterísticas, dando mayor importancia a unos atributos de calidad que a otros, asignando pesos.

En la Tabla 5 se muestra cómo se hace la composición de una subcaracterística, la Capacidad para ser analizado, a partir del conjunto de atributos que se muestran con anterioridad en la Tabla 3.

Capacidad de ser analizado	
Nivel	Táctico
Función de derivación	$An = \frac{P_{DenCC} (DenCC) + P_{DenCR} (DenCR) + P_{DenCom} (DenCom) + P_{DenDefAn} (DenDefAn)}{P_{DenCC} + P_{DenCR} + P_{DenCom} + P_{DenDefAn}}$
Atributos	<p>An: Capacidad de ser analizado del producto</p> <p>P_{DenCC} y DenCC: Peso y nivel de calidad de la densidad de la complejidad ciclomática</p> <p>P_{DenCR} y DenCR: Peso y nivel de calidad de código repetido</p> <p>P_{DenCom} y DenCom: Peso y nivel de calidad de la densidad de comentarios</p> <p>P_{DenDefAn} y DenDefAn: Peso y nivel de calidad del índice de capacidad de ser analizado</p>

Tabla 5. Capacidad de ser analizado.

Finalmente, una vez definidas las funciones para el cálculo de los indicadores de las subcaracterísticas en las que se divide la mantenibilidad, en KEMIS se define otra función para obtener el indicador de mantenibilidad, que, al igual que ocurre con el cálculo de subcaracterísticas, hace uso de pesos, dando más importancia a unas subcaracterísticas que a otras. En la Tabla 6 se muestra esta función.

Mantenibilidad	
Nivel	Estratégico
Función de derivación	$Ma = P_{An}(An) \times P_{Cam}(Cam) \times P_{Est}(Est) \times P_{Pru}(Pru)$
Atributos	<p>Ma: Mantenibilidad del producto</p> <p>P_{An} y An: Peso y nivel de calidad de la capacidad de ser analizado</p> <p>P_{Cam} y Cam: Peso y nivel de calidad de la capacidad de cambios</p> <p>P_{Est} y Est: Peso y nivel de calidad de la estabilidad</p> <p>P_{Pru} y Pru: Peso y nivel de calidad de la capacidad de pruebas</p>

Tabla 6. Mantenibilidad.

4.4. Modelos de calidad

Con todos los elementos aportados por KEMIS se genera un modelo de calidad que se almacena en una base de datos, que permite configurar los pesos de las funciones expuestas, los umbrales y las relaciones entre atributos y subcaracterísticas. Así es posible la evolución del modelo de calidad que se empleará en una organización.

A la hora de implantar KEMIS en una empresa, se debe favorecer la modificación del modelo de calidad inicial, por necesidades o restricciones propias de la empresa o, simplemente, por la evolución natural del modelo de calidad.

KEMIS permite modificar valores y relaciones de diferentes elementos del modelo de calidad: cambios en pesos, cambios en umbrales e inclusión o no de un atributo (por ejemplo, la Capacidad para ser probado) en el modelo de calidad que se va a usar, etc. En cualquier caso, un cambio en el modelo de calidad supone un cambio en los criterios de medición y, por tanto, los resultados que se obtengan seguramente serán diferentes.

Para solucionar este problema, KEMIS permite el versionado de Modelos de Calidad, de tal forma que se proporciona un mecanismo para fijar los criterios de medición (una versión concreta del modelo de calidad) con el resultado de la medición, a la vez que es posible la evolución del modelo de calidad.

Además, KEMIS posibilita la existencia conjunta de diferentes modelos de calidad, lo que permite la medición de un sistema con diferentes versiones de estos modelos. Así no sólo se facilita la evolución de los modelos de calidad, sino también su comparación.

4.5. Informes y representación

Una vez obtenida una medición basada en el modelo de calidad propuesto por la ISO/IEC 9126, se pueden obtener representaciones gráficas de diagramas de barras o incluso diagramas de Kiviati, como el de la Figura 3.

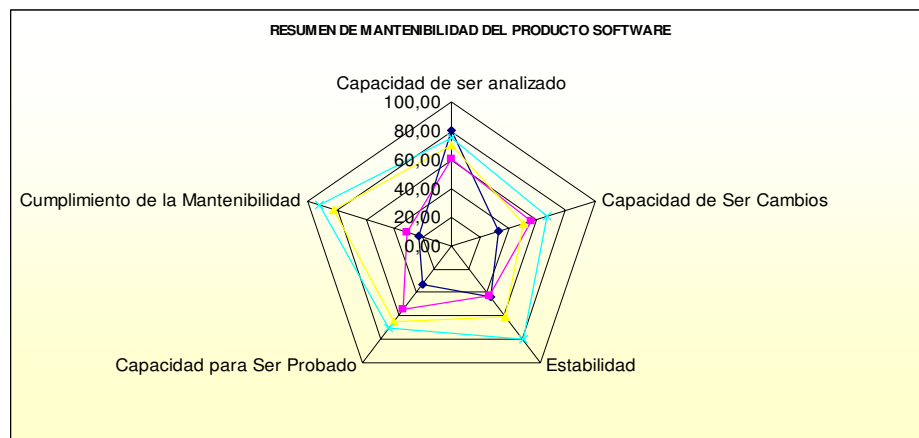


Figura 3. Kiviati de la característica de mantenibilidad de un producto.

5. Un ejemplo de evaluación

La Figura 4 muestra un ejemplo de evaluación de la calidad de varios productos de software libre disponibles en la web. Se han obtenido tres de los valores que componen la mantenibilidad: Estabilidad, Capacidad para ser cambiado y Capacidad para ser analizado. De esta manera, podemos apreciar el nivel de mantenibilidad. La columna NCSS corresponde al número de líneas de código no comentadas de cada producto.

NOMBRE	PROVEEDOR	NIVEL mantenibilidad DE PRODUCTO	NCSS	Subcaracterísticas		
				Mantenibility		
				Stability	Changeability	Analysability
jboss-maven-plugin	CodeHaus	83,05	371,00	87,26	100,00	72,47
JDepend	Clarkware Consulting	75,35	1.808,00	100,00	100,00	50,70
castor-maven-plugin	CodeHaus	72,10	263,00	98,38	100,00	45,00
ilgui	javazoom	71,81	8.538,00	85,85	50,00	75,80
quilt	apache	60,91	4.087,00	95,18	49,58	49,43
ivy	apache	60,07	24.612,00	100,00	50,00	45,13
commons-collections	apache	60,00	19.417,00	100,00	50,00	45,00
FindBugs	University of Mariland	59,85	72.195,00	99,40	50,00	45,00
JavaNCSS	Klcee	59,42	15.203,00	100,00	50,00	43,84
TightVNCviewer	TightVNC Software	56,30	3.292,00	3,80	100,00	60,70
velocity-tools	apache	52,99	4.960,00	21,98	100,00	45,00
CheckStyle	Oliver Burn	39,46	28.744,00	17,85	50,00	45,00
PMD	InfoEther	37,65	39.604,00	10,60	50,00	45,00
commons-logging	apache	34,67	1.774,00	0,00	48,69	45,00

Figura 4. Ejemplo de evaluación.

6. Conclusiones

Las normas ISO 9126 y 25000 establecen criterios para la especificación de requisitos de calidad de los productos de software, sus métricas y su evaluación.

En esta línea se desarrolla KEMIS, permitiendo la obtención de atributos de calidad y el escalado de información hacia niveles tácticos y estratégicos, basándose en un estándar, el ISO/IEC 9126. En lo que respecta a trabajos futuros, en la actualidad se están desarrollando ampliaciones del modelo que se presenta para cubrir más áreas de la ISO 25000 (funcionalidad, seguridad, etc.), destacando las pruebas de software o técnicas de caja negra. Otra línea de trabajo es la explotación de los datos recogidos por la infraestructura tecnológica, aplicando técnicas de control estadístico.

Agradecimientos

Este trabajo se ha desarrollado dentro del proyecto KEMIS, el cual es financiado por la empresa Kybele Consulting.

Referencias

- [1] Garvin, D., "What is Quality?", *Sloan Management Review*, otoño de 1984.
- [2] Humphrey, W., *Managing the Software Process*, Addison-Wesley, 1989.
- [3] INTECO, *Estudio sobre la certificación de la calidad como medio para impulsar la industria de desarrollo del software en España*, INTECO, 2008.
- [4] SEI, *Standard CMMI Appraisal Method for Process Improvement (SCAMPISM) v1.1/v1.2 Class A Appraisals Using Capability Maturity Model Integration (CMMI) v1.1 /v1.2 *2007 Year-End Update*, Software Engineering Institute, 2008.
- [5] Kitchenham, B. y Pfleeger, S. L., "Software Quality: The Elusive Target", *IEEE Software*, vol. 20, nº 1, pp. 12-21, 1996.
- [6] Maiabbaum, T.. y Wassyng, A., "A Product-Focused Approach to Software Certification", *Computer*, vol. 41, num. 2, pp. 91-93, 2008.
- [7] ISO (International Organization for Standardization), *ISO/IEC 25022 Software Engineering. Software Product Quality Requirements and Evaluation (SQuaRE). Measurement of Internal Quality*, ISO, 2005.
- [8] ISO (International Organization for Standardization), *Software Product Evaluation. Quality Characteristics and Guidelines for their Use. ISO/IEC Std 9126*, ISO, 2001.
- [9] ISO (International Organization for Standardization), *ISO 14598: 1999-2001. Information Technology. Software Product Evaluation. Parts 1-6*, ISO, 1999.
- [10] Fenton, N. y Pfleeger S., *Software Metrics: A Rigorous Approach*, Chapman & Hall, 1997.
- [11] McCabe, T., "A Software Complexity Measure", *IEEE Transactions on Software Engineering*, vol. 2, num. 4, pp. 308-320, 1976.
- [12] Hitz, M.y Montazeri, B., "Chidamber and Kemerer's metrics suite: A Measurement Theory Perspective", *IEEE Transactions on Software Engineering*, vol. 22, num. 4, pp.267-271, 1996.
- [13] Brito e Abreu, F. y Carapuça, R., "Object-Oriented Software Engineering: Measuring and Controlling the Development Process", *4th Int Conf. on Software Quality. USA*, 1994.
- [14] Erlikh, L., "Leveraging Legacy System Dollars for E-business", *IT Professional*, mayo/junio, vol. 2, nº 3, pp. 17-23, 2000.