

Introduciendo Conceptos de Metrología en el Diseño de Medidas de Software¹

Nelly Condori-Fernández¹, Oscar Pastor¹, Alain Abran², Asma Sellami²

¹Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia.
{nelly, opastor}@dsic.upv.es

²École de Technologie Supérieure ETS
1100 Notre-Dame Ouest, Montreal, Canada H3C 1K3
alain.Abran@etsmtl.ca , asma.sellami@isimsf.rnu.tn

Resumen. Una revisión del vocabulario internacional de términos de metrología fue llevada a cabo con el fin de identificar los conceptos que debieran ser considerados en el diseño de las medidas de tamaño de software. Luego, un análisis es llevado a cabo sobre el diseño de un procedimiento de medición de tamaño funcional, llamado RmFFP. Los resultados muestran que RmFFP considera la mayoría de los conceptos de metrología en comparación con otras medidas definidas en base a otros estándares.

1 Introducción

La unidad de longitud fue uno de los primeros descubrimientos de la medición hechos por el hombre, siendo el mismo cuerpo humano la base que facilitó su creación; tal como es el caso de la pulgada, la mano, el pie, y la yarda. Sin embargo, la carencia de uniformidad en estas unidades de medida, originó la necesidad por estandarizar las medidas, surgiendo la metrología como la ciencia de la medición preocupada por la definición de nuevos estándares de medición y el control de los mismos [1]. Así, con el transcurrir de los años, la metrología en constante evolución se ha ido extendiendo a nuevas áreas de tecnología, tales como los estándares eléctricos (1937), fotométricos (1937), de radiación por ionización (1960), estándares de medición en tiempo (1988) y estándares de medición en la química (2000). Sin embargo, su extensión a las tecnologías de la información aún no ha podido ser, a pesar de existir una gran cantidad de “métricas de software” en la literatura. Por consiguiente, surge la necesidad de analizar e identificar los conceptos de metrología que pudieran ser utilizados en la medición del software, ya sea por ingenieros de software interesados en el diseño de medidas de software; administradores de proyectos de software interesados en el uso coherente de estas medidas; etc.

¹ Este trabajo está subvencionado por el proyecto SESAMO con ref. TIN2007-62894 del Ministerio de Ciencia y Tecnología y parcialmente por el Programa Marie Curie del Sexto Marco de la Comunidad Europea con ref. MIF1-CT-2006-039212.

El propósito del presente artículo es en primer lugar introducir los conceptos relevantes de metrología en el diseño de las medidas de tamaño de software. Luego un análisis, en base a estos conceptos metrológicos, es ilustrado en un procedimiento de medición de tamaño funcional, llamado RmFFP [2], [3]. Este procedimiento ha sido diseñado para medir automáticamente el tamaño funcional de las especificaciones de requisitos generadas con un método basado en la transformación de modelos, OO-Method [4].

Este artículo es organizado en seis secciones. En la sección 2 se describe los trabajos relacionados a la aplicación de la metrología en la medición del software. En la sección 3, un modelo de proceso de medición es introducido para describir en mayor detalle el diseño del procedimiento de medición RmFFP. En la siguiente sección se presenta los conceptos de metrología que están relacionados con el diseño de las medidas de tamaño del software. En la sección 5, un análisis metrológico del procedimiento RmFFP es ilustrado. Finalmente, se presenta las conclusiones y trabajos futuros.

2 Metrología y Medición de Software

El término “Metrología” es definido por la Organización Internacional de Metrología Legal (OIML) [5], en el Vocabulario Internacional de Términos en Metrología (VIM) [1], como la ciencia de la medición que abarca aspectos teóricos y experimentales a cualquier nivel de incertidumbre y cualquier campo de la ciencia o la tecnología. De este modo, el alcance de la aplicación de la metrología es ilimitado, ya que las nuevas áreas de estudio incitan algunas veces a la creación de nuevas unidades de medida e instrumentos de medición. Sin embargo, el VIM que se constituye en el consenso internacional sobre la terminología de metrología, ampliamente conocido en las ciencias físicas, químicas y sociales, actualmente es casi desconocido en la comunidad de “métricas de software”.

El Instituto Nacional de Estándares y Tecnología de los EE.UU (NIST) tiene muchos años de experiencia en medir cantidades físicas, pero en lo que respecta a medición de la “tecnología de la información digital” su experiencia es relativamente nueva. El NIST identificó un conjunto de cambios, oportunidades y prioridades que surgen al intentar aplicar los conceptos de metrología a la Tecnología de la Información (TI) [6]. Ellos identificaron dos aspectos que distinguen a la “metrología de la tecnología de la información” de la “metrología física”:

Primero.- las cantidades de la TI no son medibles usando un sistema de medición físico tal como es el Sistema Internacional de Unidades. De las siete unidades base solamente el “segundo” aparece como esencial para la TI.

Segundo.- los métodos existentes para calcular expresiones de incertidumbre en la metrología física no pueden ser aplicados fácilmente en la TI. La incertidumbre puede ser mucho más difícil de definir que de cuantificar. Los métodos estadísticos sobre repetibilidad y exactitud en la metrología física no se aplican con claridad en las mediciones llevadas a cabo en las TI.

Diferentes clasificaciones de métricas de software fueron propuestas con el fin de mejorar la selección adecuada de métricas para un determinado propósito; entre las

taxonomías encontradas tenemos las de Peng y Wallace (1993) [8], Brito y Carapuca (1994) [9], Woodings (1995) [10] y Mendes et al. (2005) [11]. Sin embargo, todas estas métricas aún no han tenido una plena aceptación por los ingenieros del software. A pesar de que la ingeniería del software necesita ser soportada por medidas, métodos de medición y modelos cuantitativos y descriptivos bien probados [12].

Abran et al. afirman lo siguiente: “para que la ingeniería del software sea conocida como una disciplina de ingeniería legítima y una profesión reconocida, un consenso sobre una base del conocimiento es imperativo” [13]. Conscientes de este problema, una ontología de medición de software ha sido introducida por García et al. [14] con el fin de armonizar los diferentes estándares y modelos de medición existentes en la literatura de la ingeniería de software. Abran identificó algunas ambigüedades de alto nivel en el dominio de la medición del software y propuso un conjunto de recomendaciones para lograr una evolución de las métricas del software hacia la metrología del software [12]. Gray discutió la aplicabilidad de la Metrología a las Tecnologías de Información [15] y concluyó que no hay todavía las bases fundamentales en la medición del software que podrían ser aplicadas a los principios de la metrología; por ejemplo las líneas de código y los puntos de función tienen problemas con los principios de unidad, escala y medición de incertidumbre.

Una propuesta de Abran y Sellami introduce un modelo inicial de conceptos de medición del Vocabulario ISO de metrología [16]. Finalmente, un análisis del método estándar de medición COSMIC [17] y del reporte técnico de la ISO 9126-4 [18] fue llevado a cabo usando este modelo inicial de conceptos de metrología en [19] y [20] respectivamente.

Como el presente artículo se centra en el diseño de medidas de software, en la siguiente sección, introducimos el diseño de métodos de medición.

3 Diseño de Métodos de Medición

Jacquet y Abran definen un modelo de proceso de medición [22], identificando y definiendo un conjunto de pasos que guíen las fases de diseño, aplicación, análisis y utilización de resultados de la medición (Figura 1).



Fig. 1. Modelo de Proceso de Medición [22]

Según este modelo, el diseño de un método de medición se inicia con la *definición de objetivos*, paso donde definimos qué es lo que se desea medir, cuál es el punto vista de medición y cuál es la intención de uso. Luego *caracterizamos el concepto a ser medido*, mediante la identificación y definición del atributo a medir, (por ejemplo el tamaño, complejidad, cohesión, etc.). Teniendo en cuenta este atributo, un conjunto de conceptos, que permitan describir el software a medir, son seleccionados; este paso es llamado también *selección del meta-modelo del software*. Finalmente, un conjunto

de *reglas de asignación numérica* son definidas con el fin de cuantificar el modelo de software a medir. La Figura 2 muestra esta secuencia de pasos necesarios para llevar a cabo el diseño de un método de medición.

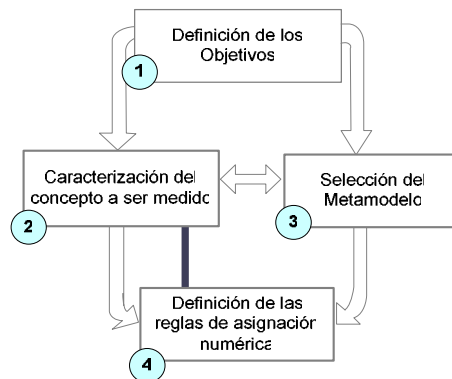


Fig. 2. Diseño de un Método de Medición [22]

A continuación, con el fin de ilustrar cada uno de los pasos de la fase de diseño del modelo de proceso de medición presentamos el diseño del procedimiento de medición RmFFP.

Definición de objetivo

Teniendo en cuenta la plantilla GQM (Goal/Question/Metric), el siguiente objetivo fue definido:

<i>Definir</i>	<i>un Procedimiento de medición</i>
<i>Para el propósito de</i>	<i>medir el tamaño funcional</i>
<i>Con respecto a las</i>	<i>especificaciones de requisitos funcionales</i>
<i>Desde el punto de vista</i>	<i>del analista</i>
<i>En el contexto del</i>	<i>Modelo de Requisitos de OO-Method.</i>

Caracterización del concepto a ser medido

A partir de nuestro objetivo, identificamos al tamaño funcional como el atributo a ser medido, el cual es definido, según el estándar ISO 14143-1[21], como el tamaño del software derivado por la cuantificación de los requisitos funcionales de usuario.

Sin embargo, esta definición, al ser muy general, necesita ser aclarada (qué entendemos por “cuantificación de los requisitos funcionales”). Para hacer esto, previamente nos apoyamos en el siguiente paso que es explicado a continuación.

Selección del metamodelo

El metamodelo a ser seleccionado debe ser acorde con el atributo a medir; en nuestro caso, como el atributo es el tamaño funcional, en la literatura existen varios metamodelos de métodos estándares candidatos, tales como IFPUG FPA, MARK II FPA, NESMA FPA y COSMIC. RmFFP fue diseñado conforme al método estándar COSMIC. Este estándar, a diferencia de los otros métodos estándares, se caracteriza por su capacidad de medir artefactos en diferentes dominios software, siendo esta una de las razones principales de nuestra elección.

El metamodelo del método estándar COSMIC fue elaborado con el fin de representar de manera más clara los diversos conceptos genéricos que son descritos en su manual de medición [25], así como también poder identificar las diferentes relaciones existentes entre dichos conceptos.

Teniendo en cuenta este conjunto relevante de conceptos, la “cuantificación de requisitos funcionales” es entendida como la cantidad de *movimientos de datos* identificados. Siendo un “*movimiento de dato*” el componente funcional básico de RmFFP.

Para facilitar la correcta identificación de este componente funcional, un conjunto de reglas fueron definidas para representar todos los conceptos del metamodelo de COSMIC en las primitivas relevantes del modelo de requisitos de OO-Method [27]. Este conjunto de reglas son presentadas detalladamente en [3] y [26]. La Tabla 1 resume esta correspondencia.

Tabla 1. COSMIC y el Modelo de Requisitos de OO-Method [3],[26]

Conceptos COSMIC		Primitivas del Modelo de Requisitos de OO-Method	
Usuarios		Actores del diagrama de casos de uso	
Frontera		Diagrama de casos de uso	
Procesos Funcionales		Casos de Uso Primarios Casos de Uso Secundarios	
Grupos de datos		Clases del diagrama de secuencia Actores diagrama de casos de uso	
Atributos de datos		Atributos de Clase	
Movimientos de Datos	Entrada	Mensaje de tipo Señal con valor Input	
	Lectura	Mensaje de tipo Consulta Condición de Mensaje Precondición de Caso de Uso Condición de relación EXTEND Restricción de integridad	
		Escritura	Mensaje de tipo Servicio con la propiedad: - New - Destroy - Update
		Salida	Mensaje de tipo Señal con valor Output

Definición de las reglas de asignación numérica

Para cuantificar el modelo de software a medir (instanciación del meta-modelo mediante la aplicación de reglas de representación previamente definidas), la *función de medición* del método estándar COSMIC fue considerado. Así mismo, con el fin de obtener el tamaño funcional de toda la especificación funcional del alcance de medición, un conjunto de reglas fue definido, las cuales fueron también presentadas en [3].

4 La Metrología en el Diseño de Medidas de Software

El VIM es un documento ISO que presenta 120 conceptos organizados en 6 categorías: cantidad y unidades, instrumentos de medición, estándares de medición, características de medición y resultados [1].

Antes de llevar a cabo el análisis metrológico de las medidas de software, primero, un modelo de los conceptos de alto nivel del VIM ha sido elaborado, Con el fin de enfatizar las relaciones identificadas entre estas categorías y facilitar su comprensión en la comunidad de ingeniería de software. Este modelo es presentado en la Figura 3.

Luego, dos preguntas fueron planteadas con la finalidad de introducir el VIM en la medición del software. Estas preguntas son:

Q1.- ¿Qué categorías del VIM debieran ser considerados durante la fase de diseño de los métodos de medición?

Q2.- ¿Qué conceptos de estas categorías debieran ser tomadas en cuenta en el diseño de las medidas de tamaño de software?

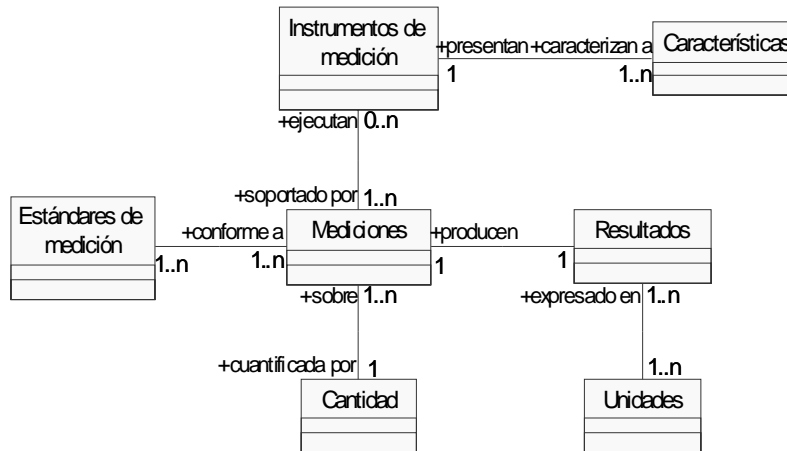


Fig. 3. Modelo de las categorías del VIM

Con respecto a la primera interrogante, partiendo del análisis inicial, llevado a cabo por Sellami y Abran [23], sobre la relación existente entre las categorías del VIM y las fases del Modelo de Proceso de Medición [21], consideramos que además de la categoría “cantidades y unidades”, las categorías “medición” y “estándares de medición” debieran ser también consideradas para la fase de diseño; ya que es necesario tener en cuenta sobre qué fundamentos de medición un método o procedimiento será diseñado. Además de que los diferentes tipos de estándares de medición debieran ser considerados a lo largo de todo el proceso de medición.. La Tabla 2 muestra la relación de las categorías del VIM que debieran ser consideradas en los respectivos pasos a seguir en el diseño de medidas de software.

Tabla 2. Relación entre categorías del VIM y los pasos de la fase de diseño

Fase de diseño según [22]	Categorías del VIM	
Paso1: Definición de objetivos		Estándares de medición
Paso2: Caracterización de concepto a ser medido	- Medición: Fundamentos-Principio de la medición	
Paso 3: Selección de meta-modelo	- Medición: Fundamentos-Método de medición - Cantidades	
Paso 4: Definición de las reglas de asignación numérica	- Cantidades y unidades	

Con respecto a la segunda interrogante, los conceptos que debemos de considerar de las categorías identificadas en la Tabla 2 son los siguientes:

- Categoría “Medición” incluye 2 sub-categorías: Fundamentos de Medición y la Medición propiamente dicha. Tal como se observa en la Tabla 2, hemos considerado únicamente los conceptos de la sub-categoría. Fundamentos de Medición: “Principio de Medición” y “Método de Medición” para el diseño de las medidas de software:
 - 1) “Principio de medición” que representa la base científica de una medición. Este concepto facilitará la caracterización del concepto o atributo a ser medido.
 - 2) “Método de medición” que representa una secuencia lógica de operaciones utilizada en una medición. Este concepto está relacionado directamente con el tercer paso de la fase de diseño, ya que el meta-modelo a seleccionar corresponde justamente al método de medición seleccionado.
- Categoría “Cantidades y unidades”, los conceptos de esta categoría están organizados en cuatro sub-categorías: sistema de cantidades, dimensión, unidades de medición y valores (Vea Figura 4).

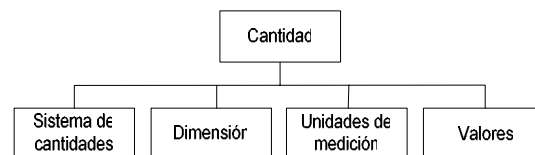


Fig. 4. Topología de alto nivel de la categoría “cantidad y unidades”

Tal como se observa en la Tabla 3, 17 conceptos están organizados dentro de estas subcategorías; de los cuales: 10 conceptos son apropiados para el análisis del diseño de medidas de tamaño. El concepto de “valor verdadero” fue considerado como indeterminado (I), debido a que no existe “medición perfecta”, más aún, en la medición del software, donde intervienen varios factores externos que podrían afectar a la obtención de este “valor verdadero” (por ejemplo: la calidad de la documentación del software a ser medido). El resto de conceptos, que son definidos en el VIM [1], no fueron considerados para este tipo de medida, la razón es que la mayoría de estos conceptos se refieren a otro tipo de cantidades y unidades que no corresponden a una medida de tamaño.

Tabla 3. “Cantidades y unidades”: Conceptos del VIM para las medidas de tamaño

.Sistema de cantidades		Unidad	
Cantidad base	√	Símbolo de unidad	√
Cantidad derivada	X	Unidad base	√
Dimensión de una cantidad		Unidad derivada	X
Dimensión	√	Unidad coherente	X
Valores		Sistema de unidades	√
Valor verdadero	I	Sistema coherente de unid.	X
Valor verdadero convencional	√	Sistema internacional de unid.	X
Valor numérico	√	Sistema Off de unidad	X
Escala de referencia convencional	√	Múltiplos de una unidad	√
		Sub múltiplos de una unidad	√

- *Categoría “Estándares de Medición”*: este concepto es entendido en el VIM como un material de referencia o sistema de medición para definir, realizar, conservar o reproducir una unidad o uno o más valores de cantidad para servir como una referencia. Tal como se observa en la Tabla 2, los estándares de medición deben ser considerados en todos los pasos de la fase del diseño.

Teniendo en cuenta los conceptos seleccionados de estas tres categorías del VIM, a continuación presentamos el análisis del diseño de RmFFP.

5 Analizando Metrológicamente el Diseño de RmFFP

En la presente sección, presentamos el análisis del diseño del procedimiento RmFFP, introducido brevemente en la sección 3, en base a las categorías: fundamentos de medición, cantidades y unidades, y estándares.

5.1 Fundamentos de medición

RmFFP ha sido diseñado conforme al *principio de medición* del método estándar ISO/IEC 19761 [17] (Vea figura 5). Por consiguiente, RmFFP si considera los fundamentos de medición que establece el VIM: principio y método. Cabe mencionar que el conjunto de operaciones generales del método de medición ha sido adaptado para facilitar la medición del tamaño funcional de las especificaciones de requisitos.

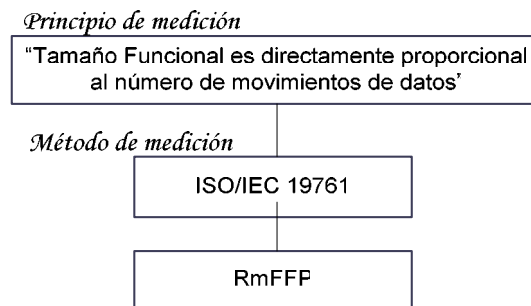


Fig. 5. RmFFP y los fundamentos de la medición

5.2 Cantidades y unidades

Teniendo en cuenta los 10 conceptos del VIM que fueron identificados como apropiados para el análisis de medidas de tamaño de software (Tabla 3), RmFFP considera lo siguiente:

En cuanto *al sistema de cantidades*, una “cantidad base” es convencionalmente aceptada como una cantidad independiente de cualquier otra cantidad. En RmFFP esta “cantidad base” es representado por el “tamaño funcional”.

En cuanto a la sub-categoría *dimensión de una cantidad*, una “dimensión” es la expresión de una cantidad representada como un producto de potencias, cuyos factores son dimensiones de *cantidades base*. Por lo tanto, la dimensión del tamaño del software estaría conformada únicamente por un factor. Sin embargo, en RmFFP, para este concepto aún falta determinar su notación dimensional.

En cuanto a la *sub-categoría valores*, el concepto “valor verdadero convencional”, en RmFFP viene a ser determinado por el valor obtenido por algún experto certificado en el estándar ISO/IEC 19761 [17],[24]. El concepto “valor numérico”, en RmFFP este valor es obtenido aplicando la “función de medición”, la misma que permite cuantificar el tamaño funcional, este valor pertenece al conjunto de números enteros positivos. El concepto “escala de referencia convencional”, en RmFFP esta escala es dada por el conjunto de valores numéricos discretos, cuyo valor mínimo es 1 movimiento de dato y su valor máximo no es establecido. Actualmente RmFFP considera que esta escala es expresada únicamente en movimientos de “grupos de dato”.

En cuanto a la sub-categoría *unidad de medición*, la “unidad base” de RmFFP es el movimiento de dato, cuyo “símbolo” es CFP (COSMIC Function Point). El concepto de “sub-múltiplo” en RmFFP actualmente no ha sido posible definirlo, ya que la relación de equivalencia entre las diversas formas de expresar un dato es difícil de determinar (grupo de dato, atributo de dato), lo que impide fraccionar la unidad de RmFFP. Sin embargo, el concepto de “múltiplo” en RmFFP si es posible determinarlo, siempre y cuando se establezca previamente algún tipo de escalonamiento por convenio. El concepto de “sistema de unidades” que incluye el conjunto de unidades base y derivadas, en RmFFP es conformado únicamente por la unidad base: movimiento de dato; a pesar de existir unidades derivadas a partir del tamaño funcional (productividad).

Por lo tanto, de los 10 conceptos de esta categoría del VIM, que debieran ser considerados en el diseño de medidas de tamaño de software, 7 han sido tomados en cuenta en el diseño de RmFFP.

5.3 Estándares de medición

Con respecto a esta categoría, el manual de medición de COSMIC [25] ha sido utilizado para el diseño de RmFFP. Sin embargo, aún hace falta la definición de otros materiales de referencia que permitan la trazabilidad de las mediciones de tamaño del software [30].

6 Conclusión

En el presente artículo se ha introducido el Vocabulario Internacional de Metrología con el fin de identificar los términos o conceptos que debieran ser considerados al momento de diseñar medidas de tamaño de software. Para esto, se ha elaborado un modelo de conceptos de alto nivel (categorías) del VIM, identificando a las categorías: medición, cantidades y unidades, y estándares de medición.

Para la *categoría medición* se ha considerado únicamente los conceptos relacionados con los *fundamentos de medición*; los demás conceptos de esta categoría (operador, procedimiento, resultados, etc.) están más implicados con la aplicación de los métodos de medición.

Para la *categoría cantidades y unidades*, alrededor del 50% de los conceptos de esta categoría fueron considerados como apropiados para el diseño de medidas de tamaño de software (10 de 17 conceptos); ya que al ser el tamaño una cantidad base, varios conceptos de esta categoría no fueron consideradas por tratarse de cantidades y unidades derivadas.

Para la *categoría estándares de medición*, se identificó la necesidad de definir “materiales de referencia” con el fin de contribuir a la trazabilidad de las mediciones de tamaño funcional y a la conservación de los actuales métodos estándar (IFPUG FPA, MARK II FPA, NESMA FPA y COSMIC).

En base a los conceptos del VIM de estas tres categorías, identificados previamente como relevantes para la medición del software, un análisis metrológico de RmFFP fue llevado a cabo; en el cual se observó que RmFFP ha sido diseñado teniendo en cuenta

la mayoría de estos conceptos. Actualmente el concepto de sub-múltiplo no ha sido todavía considerado en RmFFP, y el concepto dimensión de una cantidad tampoco ha sido contemplado ya que hace falta determinar su notación dimensional. Por lo tanto, en términos metrológicos, RmFFP resulta ser mejor en comparación con otras medidas de tamaño funcional, como los puntos de función que si tienen problemas con los principios de unidad y escala [28], [29].

Como trabajo futuro inmediato se plantea llevar a cabo un análisis del VIM para las fases de aplicación, análisis y utilización de los resultados. Además un estudio sobre la aplicación de ontologías en el análisis metrológico de las medidas de software también sería llevado a cabo.

Referencias

- [1] ISO/IEC, International Vocabulary of Basic and General Terms in Metrology (VIM), International Organization for Standardization, Geneva, Switzerland, 1993.
- [2] N. Condori-Fernández, S. Abrahão, O. Pastor, Towards a Functional Size Measure for Object-Oriented Systems from Requirements Specifications, Quality Software International Conference 2004, Braunschweig, Alemania, IEEE Computer Society, pp. 94-101.
- [3] N. Condori-Fernández, S. Abrahão, O. Pastor, On the Estimation of the Functional Size of Software from Requirements Specifications. Journal of Comput. Science and Technol., Springer, 22(3): 358-370 (2007)
- [4] O. Pastor, J. Molina, Model Driven Architecture in Practice, Springer Berlin Heidelberg, New York, 2007.
- [5] OIML: <http://www.oiml.org> última visita Diciembre 2006.
- [6] Carnahan L., Carver G., Gray M., Hogan M., Hopp T., Horlick J., Lyon G., and Messina E., Metrology for Information Technology, Standard View, 5(3):103-109, 1997.
- [7] BIPM: <http://www.bipm.org/> última visita Diciembre de 2006
- [8] Peng W., Wallace D., Software Error Analysis, NIST Special Publication, National Institute of Standards and Technology, Gaithersburg, MD, March 1993.
- [9] Brito F. and Carapua R., Candidate metrics for object oriented software within a taxonomy framework. Journal of Systems and Software, 26(1):87-96, July 1994.
- [10] Woodings, T L, A Taxonomy of Software Metrics, Software Process Improvement Network (SPIN), available from Comast Consulting, Perth, 1995.
- [11] Mendes E., Councill S., Mosley N., Towards a Taxonomy of Hypermedia and Web Application Size Metrics, International Conference on Web Engineering, Springer LNCS, Sydney, Australia, July , 2005, pp. 110-123.
- [12] Abran A., Software Metrics Need to Mature into Software Metrology (Recommendations)", presented in the NIST Workshop on Advancing Measurements and Testing for Information Technology (IT), Maryland, USA, 1998.

- [13] Abran A., Sellami A., Suryn W., Metrology, Measurement and Metrics in Software Engineering, Proceedings of the Ninth International Software Metrics Symposium, Sydney, Australia, 2003.
- [14] Garcia F., Bertoa M., Calero C., Vallecillo A., Ruiz F., Piattini M., Genero M., Towards a consistent terminology for software measurement. Journal on Information & Software Technology, Elsevier, 48(8): 631-644 (2006).
- [15] Gray M., Applicability of Metrology to Information Technology, Journal of Research of the National Institute of Standards and Technology, 4(3):103-109, 1999.
- [16] Abran A. and Sellami A., Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology, in Software Measurement and Estimation-Proceedings of the 12th International Workshop on Software Measurement, Magdeburg-Germany, October 2002, Shaker Verlag, pp. 315.
- [17] ISO, "ISO/IEC 19761: 2003, Software Engineering: COSMIC-A Functional Size Measurement Method", International Organization for Standardization-ISO, Geneva, 2003.
- [18] ISO/IEC, ISO/IEC TR 9126-4: Software Engineering – Product Quality – Part 4: Quality in Use Metrics, International Organization for Standardization, Geneva, Switzerland, 2004.
- [19] Abran A. Sellami A., Analysis of Software Measures Using Metrology Concepts – ISO 19761 Case Study, International Workshop on Software Audits and Metrics, Porto, Portugal 2004.
- [20] Abran A., E. Al-Quatish, J. Cuadrado, Investigation of the Metrology Concepts in ISO 9126 on Software Product Quality Evaluation, In Proceedings of the 10th International Conference on Computers, Athens-Greece, July 2006, pp. 864-872.
- [21] ISO, ISO/IEC 14143-1- Information Technology - Software measurement-Functional Size Measurement. Part 1: Definition of Concepts, International Organization for Standardization, Geneva, Switzerland, 1998.
- [22] Jacquet J. P. and Abran A., "From Software Metrics to Software Measurement Methods: A Process Model", in International Software Engineering Standards Symposium and Forum, ISESS 97: IEEE-Computer Society Press, 1997, pp. 128-135.
- [23] Sellami A., Abran A., The contribution of metrology concepts to understanding and clarifying a proposed framework for software measurement validation, in International Workshop on Software Measurement , Montreal , Shaker-Verlag , 2003, pp.18-40.
- [24] COSMIC: <http://www.cosmicon.com/> última visita Noviembre de 2007.
- [25] Abran A., J. M. Desharnais, S. Oligny, D. St-Pierre, and C. Symons, "COSMIC Measurement Manual – Version 3,0, The COSMIC Implementation Guide for ISO/IEC 19761:2003" École de technologie supérieure- ETS, Montreal (Canada) 2003. Available free at: <http://www.gelog.etsmtl.ca/COSMIC/>
- [26] N. Condori-Fernández, S. Abrahão, O. Pastor, El Problema de la Duplicidad de Movimientos de Datos en un Procedimiento de Medición, IDEAS 2006, Universidad de La Plata, Argentina 2006.
- [27] Insfran E., 2003. A Requirements Engineering Approach for Object-Oriented Conceptual Modeling, PhD Thesis, Valencia Polytechnic University.

- [28] Kitchenham B. Counterpoint: The Problem with Function Points, Status Report. IEEE Software, 1997, 14(2): 29-31.
- [29] Abran A., Pierre N. Function Points: A Study of Their Measurement Processes and Scale Trans-formations. Journal Systems and Software, 1994, 25(2): 171-184.
- [30] Khelifi, A., Abran, A., 'Design Steps for Development Software Measurement Standards Etalons for ISO 19761 – COSMIC-FFP', World Scientific and Engineering Academy and Society - WEAS 2007 Conference, Agios Nicolaos (Greece), July 26-28 2007, pp. 590-598.