

Revista
Española de
Innovación,
Calidad e
Ingeniería del Software



Volumen 4, Número 2 (especial X JICS), septiembre, 2008

Web de la editorial: www.ati.es/reicis

E-mail: editor-reicis@ati.es

ISSN: 1885-4486

Copyright © ATI, 2008

Ninguna parte de esta publicación puede ser reproducida, almacenada, o transmitida por ningún medio (incluyendo medios electrónicos, mecánicos, fotocopias, grabaciones o cualquier otra) para su uso o difusión públicos sin permiso previo escrito de la editorial. Uso privado autorizado sin restricciones.

Publicado por la Asociación de Técnicos de Informática

www.ati.es



Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS)

Editores

Dr. D. Luís Fernández Sanz

Departamento de Ciencias de la Computación, Universidad de Alcalá

Dr. D. Juan José Cuadrado-Gallego

Departamento de Ciencias de la Computación, Universidad de Alcalá

Miembros del Consejo Editorial

Dr. Dña. Idoia Alarcón

Depto. de Informática
Universidad Autónoma de Madrid

Dr. D. José Antonio Calvo-Manzano

Depto. de Leng y Sist. Inf. e Ing. Software
Universidad Politécnica de Madrid

Dra. Dña. Tanja Vos

Instituto Tecnológico de Informática
Universidad Politécnica de Valencia

D. Raynald Korchia

SOGETI

D. Rafael Fernández Calvo

ATI

Dr. D. Oscar Pastor

Depto. de Sist. Informáticos y Computación
Universidad Politécnica de Valencia

Dra. Dña. María Moreno

Depto. de Informática
Universidad de Salamanca

Dr. D. Javier Aroba

Depto de Ing.El. de Sist. Inf. y Automática
Universidad de Huelva

D. Antonio Rodríguez

Telelogic

Dr. D. Pablo Javier Tuya

Depto. de Informática
Universidad de Oviedo

Dra. Dña. Antonia Mas

Depto. de Informática
Universitat de les Illes Balears

Dr. D. José Ramón Hilera

Depto. de Ciencias de la Computación
Universidad de Alcalá

Contenidos

REICIS

Editorial	4
<i>Luís Fernández-Sanz, Juan J. Cuadrado-Gallego</i>	
Presentación	5
<i>Luis Fernández-Sanz</i>	
Hacia la gestión cuantitativa en la gestión de proyectos en el ámbito de las pymes	7
<i>Jose A. Calvo-Manzano, Iván García y Magdalena Arcilla</i>	
Problemas de las pymes en el nivel 2 de madurez. Una muestra sesgada	20
<i>Juan José Cukier</i>	
Mejora de procesos organizativos: análisis estadístico	33
<i>Izaskun Santamaria, Teodora Bozheva, Iñaki Martínez de Marigorta</i>	
Revisiones de código en el contexto del aseguramiento de calidad. Un caso práctico	46
<i>María José Escalona, Manuel Pérez-Pérez, O. González-Barroso, J. Ponce, J. M. Correa, A. I. Merino</i>	
Diagnóstico de la situación de la calidad del software en la industria española	58
<i>Elena Argüelles, Antonio Sepúlveda</i>	
ACCESIBILIDAD WEB: un vistazo a tres webs de administraciones públicas en España	70
<i>Jorge Sánchez, Tanja E.J. Vos</i>	
Infraestructura de pruebas para una plataforma de inteligencia de negocios: lecciones aprendidas de una experiencia académica	82
<i>Ruth Alarcón, Carla Basurto, Abraham Dávila</i>	
Perfiles del ciclo de vida del software para pequeñas empresas: los informes técnicos ISO/IEC 29110	96
<i>José A. Calvo-Manzano, Javier Garzás, Mario Piattini, Francisco J. Pino, Jesús Salillas, José Luis Sánchez</i>	
Estudio experimental de la conversión entre las unidades de medición funcional del software puntos de casos de uso e IFPUG	109
<i>Juan J. Cuadrado-Gallego, María J. Domínguez-Alda, Marian Fernández de Sevilla, Miguel Ángel Lara</i>	

Making Software Process Management Agile	122
<i>José Manuel García, José Javier Berrocal, Juan Manuel Murillo</i>	
La norma ISO/IEC 25000 y el proyecto KEMIS para su automatización con software libre	135
<i>José Marcos, Alicia Arroyo, Javier Garzás y Mario Piattini</i>	
Modelo de calidad para herramientas FLOSS que dan apoyo al modelado de procesos del negocio	148
<i>Leslibeth Pessagno, Kenyer Domínguez, Lornel Rivas, María Pérez, Luis E. Mendoza, Edumilis Méndez</i>	

Editorial

The logo for REICIS (Revista Española de Innovación, Calidad e Ingeniería del Software) is displayed in a black rectangular box. The text "REICIS" is written in a white, bold, serif font.

El grupo de Calidad del Software de ATI ha consolidado su posición como principal promotor de la disciplina de ingeniería y calidad del software con la décima edición de las Jornadas sobre Innovación y Calidad del Software (las tradicionales JICS). Estas X JICS pretenden además potenciar la presencia iberoamericana en este foro de promoción de la cultura de la calidad del software y de la innovación en el desarrollo de sistemas y aplicaciones por lo que constituyen la promoción de una I Conferencia Iberoamericana de Calidad del Software (CICS). Por otra parte, las X JICS incorporan la presencia de la ponencia de un destacado experto europeo en la disciplina de ingeniería de software como es Darren Dalcher, Director del UK National Centre for Project Management en la Middlesex University y editor de la revista Software Process Improvement and Practice.

Por otra parte, queremos resaltar la línea de calidad de los trabajos, eminentemente prácticos pero rigurosos, aceptados entre los remitidos en la convocatoria de contribuciones: las ponencias aceptadas (con una tasa de rechazo del 40%) han sido sometidos a un completo proceso de revisión por el comité de programa así como a una cuidadosa labor de revisión de estilo, de terminología y de ortotipografía para garantizar el mejor resultado para nuestros lectores. Por supuesto, no cabe olvidar el apoyo de los patrocinadores (Telelogic, Steria, Deiser, GESEIN y SOGETI) no sólo aportando recursos sino también interesantes presentaciones de experiencias prácticas de sus expertos. Los debates promovidos en las mesas redondas así como la promoción de las actividades de comunicación y *networking* entre los participantes, tanto a nivel presencial como a través de la lista de distribución, los medios electrónicos y la nueva oferta formativa con plataforma *e-learning*. En definitiva, el evento más completo con toda la información disponible en la página del grupo de Calidad del Software (www.ati.es/gtcalidadsoft) acorde a la trayectoria pionera en España que, desde 1997, está proporcionando, a través de la Asociación de Técnicos de Informática, el apoyo para la productividad y la calidad en los proyectos de software. Este perfil ha sido reconocido por el apoyo del Ministerio de Industria, Turismo y Comercio con su apoyo institucional dentro de la convocatoria de la orden ITC/390/2007. Por último, debemos resaltar la aportación de datos de gran importancia no sólo mediante los eventos organizados sino también a través de la realización de estudios específicos (por ejemplo, sobre las prácticas de pruebas, el diseño de casos y los factores que dificultan su implantación eficiente y eficaz en las organizaciones) que permiten un mejor conocimiento de la práctica real de esta disciplina en España.

Luis Fernández Sanz
Juan J. Cuadrado-Gallego
Editores

En este número especial de septiembre de 2008 de REICIS, por primera vez en la historia de nuestra revista, esta publicación se convierte en el vehículo de difusión del evento decano en España en el ámbito de la ingeniería y la calidad del software: las Jornadas de Innovación y Calidad del Software (JICS) que alcanzan así su décima edición desde su inicio en 1998. En esta ocasión, el Grupo de Calidad del Software de ATI (www.ati.es/gtcalidadsoft) no sólo ha querido cumplir con esta decena de ediciones sino que ha apostado por una apertura a nuevos retos como la presencia de eminentes ponentes invitados de gran presencia internacional y la potenciación de los vínculos iberoamericanos para convertir a este evento en la referencia sobre calidad del software en la amplia comunidad latina. Los trabajos aceptados han sido sometidos a un completo proceso de revisión por el comité de programa así como a una cuidadosa labor de revisión de estilo, terminología y ortotipografía para garantizar la mejor calidad para nuestros lectores. Este número especial constituye en definitiva la publicación de las actas de las X JICS y, por ello, cuenta con un tamaño mayor del habitual. Esperamos repetir este número especial el próximo año con la undécima edición de las Jornadas de Innovación y Calidad del Software. Agradecemos la labor del comité de programa coordinado por la Dr. M. Idoia Alarcón (Universidad Autónoma de Madrid) y compuesto por la siguiente lista de expertos:

- Antonia Mas (Universitat de les Illes Balears)
- Luis de Salvador (AGPD)
- Ricardo Vargas (Universidad del Valle de Méjico)
- Javier Tuya (Universidad de Oviedo)
- Antonio de Amescua (Universidad Carlos III de Madrid)
- María Moreno (Universidad de Salamanca)
- José Antonio Calvo-Manzano (Universidad Politécnica de Madrid)
- José Antonio Gutiérrez de Mesa (Universidad de Alcalá)
- Isabel Ramos (Universidad de Sevilla)
- Esperança Amengual (Universitat de les Illes Balears)
- José Ramón Hilera (Universidad de Alcalá)
- Mercedes Ruiz (Universidad de Cádiz)
- María Teresa Villalba (Universidad Europea de Madrid)
- Adolfo Vázquez (INSA)
- María José Escalona (Universidad de Sevilla)
- Ana Araújo (Ministerio de Medio Ambiente)
- Antonio Rodríguez (Telelogic)
- Gurutze Miguel (TQS)
- Beatriz Pérez (Centro de Ensayos de Software, Uruguay)
- José Javier Martínez (Universidad de Alcalá)
- José Díaz (SSQTB)

Luis Fernández Sanz

Modelo de calidad para herramientas FLOSS que dan apoyo al modelado de procesos del negocio

Leslibeth Pessagno, Kenyer Domínguez, Lornel Rivas,
María Pérez, Luis E. Mendoza, Edumilis Méndez

Laboratorio de Investigación en Sistemas de Información (LISI).
Departamento de Procesos y Sistemas. Universidad Simón Bolívar
leslibethpessagno@gmail.com
{kdoming, lrivas, movalles, lmendoza, emendez}@usb.ve

Abstract

The selection of FLOSS tools supporting business modeling discipline is a complicated task; besides verifying the proper use of language and notations such as BPMN [1], UML [2] and SPEM, we must validate that such tools meet the features of this type of software. Consequently, the quality of these tools should be assessed through a quality model that allows for determining fulfillment of such requirements. This research, currently in progress, included the use and creation of a Quality Systemic Model (MOSCA) [3]. This model is based on ISO 9126 [4], Dromey's [5] quality model, and the Goal/Question/Metric paradigm [6]. In addition, the instantiation proposed includes software attributes relating to functionality, usability, and maintainability and establishes 75 new metrics, for a total of 128 metrics, to assessing FLOSS tools for Business Modeling purposes. Four tools were selected in order to validate this proposal, namely Eclipse Process Framework Composer (EPFC) [7], StarUML [8], Intalio [9] and y DIA (<http://www.gnome.org/projects/dia/>), the chosen tool being EPFC, the highest quality tool to support this discipline.

Key words: business modeling, software engineering tools, FLOSS, ISO 9126, software quality model.

Resumen

Seleccionar herramientas FLOSS que dan apoyo a la disciplina modelado del negocio es una tarea complicada ya que, además de verificar la correcta utilización de lenguajes y notaciones como BPMN [1], UML [2] y SPEM, se debe validar que satisfacen las propiedades que caracterizan a este tipo de software. Por tanto, es necesario evaluar la calidad de estas herramientas a través de un modelo de calidad que permita determinar el cumplimiento de dichos requisitos. En esta investigación en progreso se utilizó y se realizó un instanciación del modelo sistémico de calidad (MOSCA) [3]. Este modelo se basa en los estándares de ISO 9126 [4], en el modelo de calidad de Dromey [5] y en el paradigma de objetivos, preguntas y métricas [6]. La instanciación propuesta incluye atributos del software relacionados con la funcionalidad, la usabilidad y la mantenibilidad, estableciendo 75 nuevas métricas para un total de 128 que permiten evaluar herramientas FLOSS para el modelado del negocio. Con el fin de validar esta propuesta se seleccionaron cuatro

herramientas: Eclipse Process Framework Composer (EPFC) [7], StarUML [8], Intalio [9] y DIA (<http://www.gnome.org/projects/dia/>). EPFC se escogió como la herramienta con más alto nivel de calidad en cuanto a apoyo a esta disciplina.

Palabras clave: modelado del negocio, herramientas de ingeniería del software, FLOSS, ISO 9126, modelo de calidad de software.

1. Introducción

La versatilidad de las herramientas de ingeniería del software que dan apoyo al modelado del negocio basadas en software libre, su amplia variedad en el mercado y el diverso nivel de funcionalidad que ofrecen, hacen compleja su selección. Además, existen distintos lenguajes de modelado que permiten representar procesos del negocio —entre ellos, EPM [10] y BPMN [1]—; incluso existen algunos especializados solo en procesos de desarrollo de software, como el perfil de negocios de UML [2], SPEM y la extensión de Eriksson y Penker [11], lo cual hace aún más complicada la evaluación de este tipo de herramientas.

En el mercado existen herramientas que dan apoyo a los lenguajes mencionados antes. Algunas de ellas se caracterizan por combinar las características de software libre [12] y Software Open Source [13], que se pueden resumir en los siguientes aspectos: acceso al código fuente, modificación del código, sin restricciones de uso, copia y re-distribución [12, 15, 16]. No obstante, seleccionar este tipo de herramientas no es una tarea fácil, pues deben satisfacer, además de los requisitos de modelado de procesos del negocio, las propiedades de Free/Libre Open Source Software (FLOSS) [14].

Por esta razón, la presente investigación pretende alcanzar los siguientes objetivos: (a) propuesta de una instanciación del modelo sistémico de calidad (MOSCA) [3] que permita evaluar la calidad de herramientas FLOSS que den apoyo al modelado del negocio; (b) evaluación de la utilidad del modelo aplicándolo a la estimación de la calidad de un conjunto de herramientas con estas características.

En la bibliografía existente se han encontrado dos trabajos relacionados con el que se presenta en este artículo. Por un lado está el método de cualificación y selección de Software Open Source [15] que, debido a sus características, requiere considerar criterios asociados a la modificación, licencias y madurez de software, entre otros. Y por otro lado, existe un meta-modelo para evaluar lenguajes de modelado de procesos del negocio [16] que permite representar un amplio rango de conceptos de procesos del negocio. Para

garantizar su cobertura, el meta-modelo está conformado por cinco perspectivas: funcional, organizativa, comportamiento, información y contexto de procesos del negocio. Sin embargo, ninguna de estas dos iniciativas está orientada a establecer un modelo sistémico de calidad para herramientas FLOSS que dan apoyo al modelado de procesos del negocio.

Este trabajo consta de siete secciones. En esta se presenta la introducción y seguidamente, la metodología empleada en la investigación. El epígrafe tercero describe MOSCA; el cuarto presenta el modelo de calidad propuesto para herramientas FLOSS que dan apoyo al modelado del negocio. Las secciones quinta y sexta describen la aplicación del modelo y los resultados obtenidos. Por último, en la sección séptima se presentan las conclusiones y recomendaciones.

2. Metodología

En este trabajo se utilizó el Framework Metodológico Sistémico para investigar sistemas de información [17], el cual se basa en el método de investigación-acción [18] y en la metodología DESMET [19]. El primero se desarrolla en cinco fases: diagnosticar, planificar la acción, realizar la acción, evaluar y especificar el aprendizaje [17], mientras que la metodología DESMET interviene para complementar la evaluación del modelo. Propone nueve métodos de evaluación, de los cuales se empleó el método de análisis de las características por estudio de caso [17]. Además, se incluyó el enfoque basado en objetivos, preguntas y métricas, con el propósito de hacer posible la medición del software en un contexto de mejora de la calidad [6].

3. Modelo de calidad sistémico (MOSCA)

El modelo sistémico de calidad se basa en la matriz de calidad global sistémica de Callaos e integra tres modelos de calidad [20, 21]: producto, proceso de desarrollo y perspectiva humana. MOSCA está constituido por estos niveles:

- Nivel 0. Dimensiones. Aspectos internos y contextuales del producto, el proceso y la perspectiva humana.
- Nivel 1. Categorías. Se contemplan 14 categorías, cinco pertenecientes al proceso, seis pertenecientes al producto y tres para la perspectiva humana.
- Nivel 2. Características. Cada categoría tiene asociado un conjunto de

características que definen las áreas claves para lograr, controlar y asegurar la calidad en las tres perspectivas. Hay 56 características asociadas para el producto, 27 para la perspectiva del proceso y 15 para la parte humana.

- Nivel 3. Métricas, usadas para medir la calidad sistémica. Existen 715 métricas.

MOSCA evalúa el producto según normas internacionales, pues las categorías presentadas antes coinciden con las características del estándar ISO 9126 [4], que son establecidas para garantizar la calidad de producto de software.

Para la aplicación de MOSCA existe el siguiente algoritmo:

1. Estimación de la calidad del producto. Inicialmente, se debe medir la categoría de funcionalidad del producto. Si esta cumple con el 75% de las características necesarias propuestas para esta categoría, se prosigue con el siguiente paso.
2. Instanciación del submodelo del producto. De las cinco categorías restantes, se seleccionan dos. El algoritmo recomienda trabajar con un máximo de tres categorías, pues si se seleccionan más, podrían presentar conflictos.
3. Estimación de la calidad para cada categoría. Para las dos categorías seleccionadas en el paso anterior se debe: (a) aplicar las métricas propuestas en el submodelo del producto para las categorías seleccionadas; (b) verificar que el 75% de las métricas están dentro de los valores óptimos (mayor o igual a tres) para cada una de sus características; y (c) evaluar la categoría. Para que una categoría sea satisfecha, al menos el 75% de sus características deben ser altamente satisfechas. Esto garantiza coherencia y consistencia con los niveles de aceptación establecidos por el modelo.
4. Estimación de la calidad del producto partiendo de las categorías evaluadas. Si no se satisface la categoría funcionalidad, el algoritmo finaliza y la calidad del producto de software será nula. Si un producto cumple con los objetivos para los cuales se creó (funcionalidad), tendrá una calidad básica. Si satisface solo una de las categorías seleccionadas, además de la funcionalidad, tendrá un nivel de calidad intermedio; si satisface todas las categorías seleccionadas, tendrá un nivel avanzado.

4. Modelo de calidad propuesto para herramientas FLOSS que dan apoyo al modelado de procesos del negocio

De acuerdo con el algoritmo de MOSCA, para el proceso de evaluación se deben elegir tres

de las seis categorías de la dimensión producto, entre las cuales ha de incluirse la funcionalidad. Las otras dos categorías seleccionadas son la usabilidad (pues las herramientas de modelado deben ser fáciles de comprender y utilizar) y la mantenibilidad. Dado que el alcance de esta investigación cubre herramientas FLOSS, además estas categorías también son propuestas en [15] entre sus criterios de evaluación. Para cada una de estas categorías se seleccionó un subconjunto de características y para estas, un subconjunto de métricas; en algunos casos fue necesario añadir nuevas métricas. A continuación se expone en qué consiste cada una de estas categorías, así como los criterios utilizados para seleccionar sus características.

4.1. Funcionalidad

La funcionalidad es la capacidad del producto de software para proveer funciones que cumplan con necesidades específicas o implícitas cuando se usa en unas condiciones determinadas [4]. Es una categoría fundamental porque se espera que todo producto cumpla con los propósitos para los que fue creado; en este caso, es necesario que la herramienta que se va a evaluar dé apoyo, al menos, a los lenguajes UML, SPEM, BPMN y EPM. De las ocho características de esta categoría se escogieron las descritas en la Tabla 1.

Categoría	Definición y justificación
Ajuste a los propósitos (FUN 1)	Es la capacidad del producto de software para proveer un conjunto de funciones apropiado según tareas y objetivos específicos del usuario. En este trabajo se requiere que una herramienta cuente con funcionalidades que permitan representar modelos de procesos del negocio.
Precisión (FUN 2)	Es la capacidad del producto de software para proveer los resultados correctos. La herramienta que dé apoyo a lenguajes para modelado del negocio debe proveer la notación adecuada para la representación de los procesos que lo conforman.
Interoperatividad (FUN 3)	Es la capacidad del producto de software para interactuar con uno o más sistemas especificados. Dado que la herramienta permitirá representar modelos de procesos del negocio y que estos pueden estar compuestos por uno o más de ellos, hay que saber si el producto tiene funcionalidades utilizadas por otro sistema o si otros sistemas utilizan sus funcionalidades para llevar a cabo el modelado.
Corrección (FUN 5)	Esta característica se divide en tres categorías relacionadas con la capacidad de cómputo, completitud y consistencia. Alguna violación de una de estas propiedades puede significar que el software no tenga la funcionalidad esperada. Es necesario que los diagramas generados por la herramienta sean completos y consistentes.
Encapsulado (FUN 7)	Las variables, las constantes y los tipos se deben usar en el contexto en el que son definidos. El modo en que se usen puede tener un impacto significativo sobre la modularidad y, por lo tanto, sobre la calidad de los módulos y programas.

Tabla 1. Características para la categoría funcionalidad.

A fin de de cumplir con los requisitos necesarios para representar modelos de procesos de negocio, se añadieron las subcaracterísticas presentadas en la Tabla 2.

Subcaracterística	Descripción
Diagramas (FUN 1.1)	Es la capacidad de la herramienta para llevar a cabo la representación de modelos de procesos de negocio a través de los diagramas asociados a un lenguaje para el modelado del negocio, como BPMN, EPM, Extensión Eriksson & Penker, SPEM o el perfil UML para negocios.
Documentación (FUN 1.2)	Es la capacidad de la herramienta de proveer mecanismos para la generación de documentación, así como para permitir realizar anotaciones en los diagramas.
Lenguajes (FUN 1.3)	Se refiere a los lenguajes empleados para representar modelos de procesos de negocio que la herramienta es capaz de proveer al usuario.
Detalles de abstracción (FUN 2.1)	Es la capacidad de la herramienta de modelado para detallar las abstracciones que representan un modelo de procesos del negocio.
Completo (FUN 5.1)	Es la capacidad de la herramienta de proveer todos los estereotipos, conectores, símbolos, etc., pertenecientes al lenguaje de modelado. También permite verificar si esta posee todos los diagramas asociados al lenguaje.
Consistente (FUN 5.2)	Es la capacidad de verificar si los símbolos provistos por la herramienta corresponden a los del lenguaje de modelado al cual pertenecen y si estos no pueden ser mezclados o intercambiados entre lenguajes.
Taxonomía (FUN 7.1)	Es la capacidad de organizar los diagramas y lenguajes a través de alguna clasificación.

Tabla 2. Subcaracterísticas propuestas para la categoría funcionalidad de herramientas FLOSS que dan apoyo al modelado del negocio.

4.2. Usabilidad

La usabilidad es la capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario en unas condiciones específicas [4]. Las herramientas de modelado deben permitir realizar la diagramación de forma sencilla, y han de presentar las funcionalidades de forma accesible y consistente, para que los distintos usuarios que las usen trabajen cómodamente, reduciendo el tiempo y esfuerzo de aprendizaje [22]. Por ello, de las 11 características de esta categoría se seleccionaron las cuatro descritas en la Tabla 3.

Categoría	Definición y justificación
Facilidad de comprensión (USA 1)	Es la capacidad del producto de software para facilitar al usuario entender el software y la forma en que se puede usar para efectuar diferentes tareas bajo condiciones específicas. Es fundamental que las funciones asociadas a la diagramación sean fáciles de comprender y ubicar por los diferentes tipos de usuarios (programadores, analistas, etc.).
Interfaz gráfica (USA 3)	Se refiere a la capacidad del producto de software para captar la atención del usuario. La interfaz gráfica debe agradar visualmente al usuario, además de presentar consistentemente la información y las acciones de que dispone la herramienta.
Operatividad (USA 4)	Es la capacidad del producto de software para facilitar al usuario su uso y control. Es importante que la herramienta requiera el mínimo esfuerzo para su utilización, pues el proceso de diagramación debe sencillo y rápido de realizar.
Autodescriptivo (USA 11)	Una forma estructural es autodescriptiva si su propósito es evidente en el nombre de los módulos y los identificadores tienen significado en el contexto de la aplicación. Es necesario considerar esta característica, pues si la herramienta es autodescriptiva, se podrá operar fácilmente con ella.

Tabla 3. Características para la categoría usabilidad.

La Tabla 4 muestra las subcaracterísticas añadidas al modelo, así como su descripción.

Característica	Subcaracterística	Descripción
Facilidad de comprensión (USA 1)	Ergonomía (USA 1.1)	Es la capacidad que tiene la interfaz de la herramienta para facilitar la interacción hombre-máquina.
Operatividad (USA 4)	Control de errores (USA 4.1)	Se refiere a los mecanismos que provee la herramienta para que el usuario pueda recuperarse de errores.
	Documentación (USA 4.2)	Se trata de la documentación de las funcionalidades provistas por la herramienta.

Tabla 4. Subcaracterísticas y métricas propuestas para la categoría usabilidad para las herramientas FLOSS que dan apoyo al modelado del negocio.

4.3. Mantenibilidad

Mantenibilidad es la capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software ante cambios del ambiente, en requisitos y especificaciones funcionales [4]. Las herramientas desarrolladas en FLOSS, a diferencia de las de software propietario, deben prestar especial atención a aspectos relacionados con la visibilidad, reutilización y modificación. Por esta razón, de las catorce características asociadas a esta categoría, se seleccionaron las descritas en la Tabla 5.

Categoría	Definición y justificación
Capacidad de análisis (MAB 1)	Es la capacidad del producto de software para ser diagnosticado por deficiencias o causas de fallos en el software, o por partes que hay que modificar. Para poder evaluar la herramienta y determinar cuáles serán las mejoras que pueden llevarse a cabo, es necesario que esta permita realizar fácilmente el diagnóstico.
Capacidad de cambio (MAB 2)	Es la capacidad del producto de software para facilitar la implementación de una modificación específica. Es conveniente que una herramienta FLOSS posibilite a la comunidad de software libre a realizar aportes que permitan incurrir en potenciales mejoras.
Estabilidad (MAB 3)	Es la capacidad del producto de software para evitar efectos inesperados después de modificaciones en el software. Se requiere que, al modificar la herramienta, no se generen efectos colaterales en funcionalidades que trabajaban correctamente.
Acoplamiento (MAB 5)	Es una medida de la interconexión entre los módulos de una estructura de programa. En el diseño de software se intenta conseguir el menor nivel posible de acoplamiento, ya que facilita la modificación del software. Ocurre todo lo contrario cuando el diseño presenta un acoplamiento fuerte.
Cohesión (MAB 6)	Una forma estructural es cohesiva si todos sus elementos están enlazados estrechamente unos a otros y si contribuyen a llevar a cabo un simple objetivo o función. Una alta cohesión en herramientas FLOSS facilitaría su reutilización y evitaría la aparición de efectos colaterales durante la modificación del software.
Atributos de madurez del software (MAB 8)	Son el conjunto de las características físicas y medidas asociadas a la edad y uso del sistema de software objetivo. Una herramienta debe ser lo suficientemente madura para aumentar las posibilidades de éxito en las modificaciones realizadas.

Tabla 5. Características para la categoría mantenibilidad.

La Tabla 6 presenta las subcaracterísticas que se añadieron a la categoría

mantenibilidad.

Característica	Subcaracterística	Descripción
Capacidad de análisis (MAB 1)	Legibilidad del código (MAB 1.1)	Se refiere a la capacidad del código fuente de ser leído por cualquier desarrollador, incluso por aquel que no pertenezca al proyecto.
	Licencia (MAB 2.1)	Son las propiedades que tiene la licencia de la herramienta. Entre otras cosas, la licencia debe garantizar las libertades del usuario para acceder al código fuente, modificarlo, copiarlo o distribuirlo
Capacidad de cambio (MAB 2)	Modificación (MAB 2.2)	Es la capacidad que tiene el código fuente de la herramienta para ser modificado.
	Documentación (MAB 2.3)	Se refiere a la documentación técnica del software: documentación del código fuente, de diseño, plan de riesgos, etc.
Estabilidad (MAB 3)	Servicios (MAB 3.1)	Se refiere a los servicios de soporte, entrenamiento y consultoría provistos por la comunidad de desarrollo de la herramienta para respaldar a los usuarios.
Atributos de madurez (MAB 8)	Adopción (MAB 8.1)	Permite medir el grado de aceptación de la herramienta en el mercado, tanto por personas física, como por empresas y organizaciones.

Tabla 6. Subcaracterísticas propuestas para la categoría mantenibilidad para herramientas FLOSS que dan apoyo al modelado del negocio.

La instanciación de MOSCA para herramientas FLOSS para el modelado del negocio presenta 128 métricas, de las cuales 75 son métricas nuevas (42 en funcionalidad, nueve en usabilidad y 24 en mantenibilidad). La Figura 1 muestra el subárbol de la adaptación.

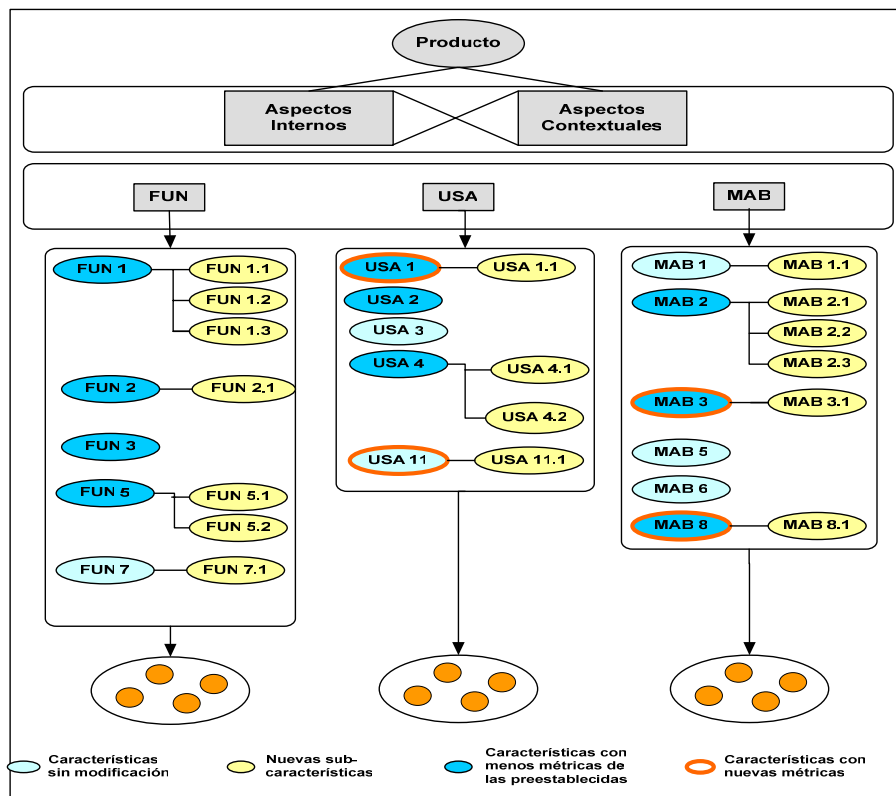


Figura 1. Instanciación de MOSCA para herramientas FLOSS que dan apoyo al modelado del negocio.

5. Aplicación de la instanciación

La instanciación de MOSCA se aplicó a cuatro herramientas: Eclipse Process Framework Composer (EPFC) [7], StarUML [8], Intalio Designer [9] y DIA. Todas las herramientas permiten modelar procesos del negocio utilizando lenguajes como BPMN [1], UML (perfil de negocios [2]), SPEM y EPM [10]. EPFC, además de ser una herramienta empleada para representar diagramas, permite gestionar procesos. Sin embargo, para este trabajo de investigación es de interés estudiar un subconjunto de la parte funcional, específicamente, el modelado visual de procesos del negocio. Los resultados obtenidos en la evaluación se detallan en la Tabla 7.

En total se invirtieron aproximadamente 88 horas para la aplicación de la instanciación. El proceso se realizó de la siguiente forma: se descargaron del site de la comunidad de desarrollo correspondiente las últimas versiones de las herramientas y luego se instalaron (una hora por herramienta). Posteriormente, se ejecutaron las aplicaciones una a una para iniciar el proceso de familiarización, cuyo tiempo estimado fue de cuatro horas por herramienta; después se verificó el cumplimiento o no de las métricas de funcionalidad (tres horas por herramienta) y usabilidad (una hora por herramienta). Para la evaluación de las métricas de mantenibilidad se revisó la información y documentación existente en los sites oficiales de las herramientas (siete horas por herramienta).

Categoría	Característica	Subcaracterística	EPFC	StarUML	Intalio	DIA
Funcionalidad	Ajuste a los propósitos	Diagramas	100,00	82,35	100,00	55,56
		Documentación	100,00	100,00	60,00	60,00
		Lenguajes	50,00	75,00	60,00	55,00
	Precisión	Detalles de abstracción	100,00	70,00	60,00	50,00
	Interoperatividad	Interoperatividad	46,67	20,00	20,00	20,00
	Corrección	Completo	76,36	93,84	100,00	93,33
		Consistente	60,00	66,67	100,00	46,67
	Encapsulado	Taxonomía	100,00	46,67	-	-
Porcentaje de satisfacción			77,78	78,69	77,14	64,67
Usabilidad	Facilidad de comprensión	Facilidad de comprensión	64,00	80,00	84,00	92,00
		Ergonomía	100,00	100,00	100,00	100,00
	Capacidad de aprendizaje	Capacidad de aprendizaje	80,00	80,00	80,00	20,00
	Interfaz gráfica	Interfaz gráfica	100,00	94,29	94,29	65,71
	Operatividad	Operatividad	86,00	72,00	52,00	64,00
		Control de errores	100,00	100,00	100,00	100,00
		Documentación	80,00	96,00	64,00	84,00
	Autodescripción	Autodescripción	100,00	100,00	90,00	100,00
Porcentaje de satisfacción			86,25	85,63	74,38	75,00

Mantenibilidad	Capacidad de análisis	Capacidad de análisis	100,00	80,00	30,00	100,00
		Legibilidad del código	100,00	100,00	20,00	100,00
	Capacidad de cambio	Capacidad de cambio	100,00	68,00	28,00	72,00
		Licencia	100,00	100,00	77,14	100,00
		Modificación	100,00	53,33	33,33	80,00
		Documentación	100,00	60,00	20,00	20,00
	Estabilidad	Estabilidad	73,33	20,00	20,00	20,00
		Servicios	100,00	20,00	100,00	60,00
	Acoplamiento	Acoplamiento	100,00	100,00	20,00	60,00
	Cohesión	Cohesión	100,00	100,00	20,00	80,00
	Atributos de madurez del software	Atributos de madurez del software	84,44	66,67	44,44	75,56
		Adopción	84,00	68,00	20,00	36,00
	Porcentaje de satisfacción			93,48	70,43	40,00
Nivel de calidad			Avanzada	Media	Básica	Nula

Tabla 7. Resultados de la evaluación.

6. Resultado del análisis

La Figura 2 muestra los resultados obtenidos en cada categoría. En funcionalidad, EPFC obtuvo un 77,78%; StarUML, un 78,69%; Intalio, un 77,14% y DIA, un 64,67% de métricas satisfechas. De acuerdo con el algoritmo de MOSCA, DIA no alcanzó el nivel mínimo de aceptación requerido, que es el 75%. También se observa que, en cuanto a los resultados asociados a la categoría usabilidad, EPFC obtuvo la puntuación más alta, con el 86,25%; el segundo y el tercer lugar lo ocupan StarUML y DIA, respectivamente. Intalio no alcanzó el porcentaje mínimo requerido. Finalmente, en la categoría mantenibilidad, la única herramienta que con una puntuación mayor al 75% es EPFC, con un 93,48%.

DIA no satisfizo el porcentaje mínimo requerido en la funcionalidad. Esto la califica como una herramienta con un nivel de calidad nulo. Por su parte, Intalio obtuvo una puntuación mayor al 75% únicamente en la categoría de funcionalidad, por lo que tiene un nivel de calidad básico. StarUML satisface dos de sus categorías: funcionalidad y usabilidad, por lo que posee un nivel de calidad intermedio.

Finalmente, la única herramienta que satisface las tres categorías seleccionadas para la instanciación, con una puntuación superior al 75% es EPFC. En consecuencia, presenta un nivel de calidad avanzado. Por esta razón, EPFC es la herramienta seleccionada para realizar las mejoras relacionadas con las características que sean pertinentes y factibles para la investigación. La presentación de estos resultados escapa del alcance de este artículo y

serán publicados en el futuro.

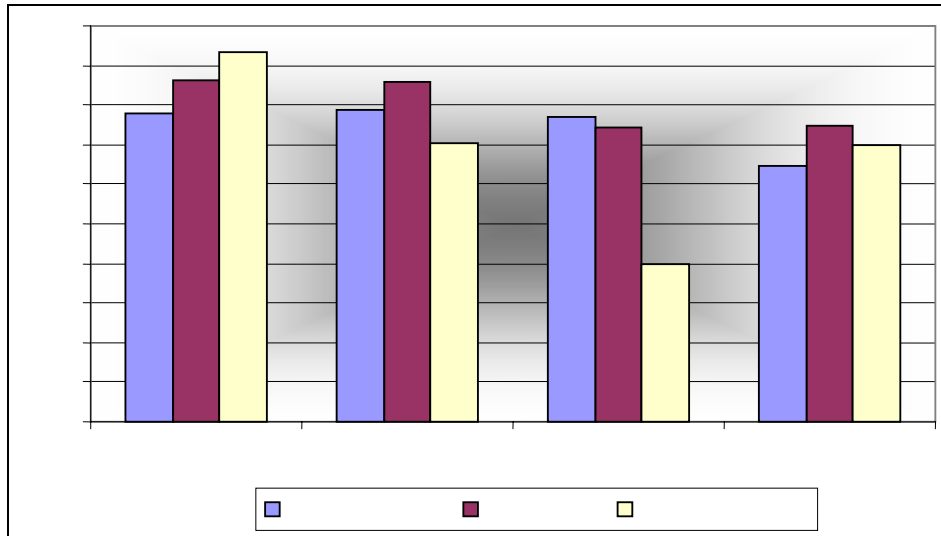


Figura 2. Resultados de las categorías funcionalidad, usabilidad y mantenibilidad para cada herramienta.

7. Conclusiones y recomendaciones

Este trabajo propone una instanciación de MOSCA que permite evaluar la calidad de herramientas de ingeniería del software basadas en FLOSS, que den apoyo al modelado del negocio, las cuales, además de cubrir las funcionalidades requeridas, deben ser fáciles de utilizar y ser adaptables al cambio. Este modelo fue aplicado a EPFC [7], StarUML [8], Intalio [9] y DIA, con la finalidad de comprobar la utilidad del modelo y, además, seleccionar la herramienta con un nivel de calidad avanzado para mejorarla en un futuro. En este caso, la seleccionada fue EPFC. Los resultados de esta investigación pueden servir de guía para pequeñas y medianas empresas que precisen seleccionar una herramienta de apoyo a esta disciplina. Siendo EPFC una herramienta FLOSS, puede resultar de fácil acceso para estas organizaciones. Finalmente, se debe indicar que MOSCA puede ser adaptado para herramientas de modelado del negocio con características específicas. Por esta razón, para futuros trabajos de investigación se recomienda evaluar herramientas que den apoyo a la gestión de procesos del negocio.

Agradecimientos

Esta investigación ha sido financiada por el Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT) de la República Bolivariana de Venezuela, a través del proyecto G-2005000165.

Referencias

- [1] Business Process Management Initiative (BPMI), *Business Process Modeling Notation Specification BPMN 1.0*, OMG, 2006.
- [2] Johnston, S., *Rational UML Profile for Business Process*, IBM, 2004.
- [3] Mendoza, L., Pérez, M. y Grimán, A., “Prototipo de modelo sistémico de calidad (MOSCA) del software”, *Computación y Sistemas*, vol. 8, nº 3, pp. 196-217, 2005.
- [4] ISO, *ISO/IEC 9126-1, Software Engineering. Product Quality. Part 1: Quality Model*, ISO, 2001.
- [5] Dromey, G., “A Model for Software Product Quality”, *IEEE Transactions on Software Engineering*, vol. 21, nº 2, pp. 146-162, 1995.
- [6] Basili, V., Caldiera, G. y Rombach, H., “The Goal Question Metric Approach”, en: Marciniak, J. J. (ed.), *Encyclopedia of Software Engineering*, Wiley, pp. 528–532, 2001.
- [7] Eclipse Process Framework Project (EPF), *Documentación del proyecto*, EPF, 2006. <http://www.eclipse.org/epf/> (consultado en marzo de 2008)
- [8] StarUML, *UML2.0 Modeling Tool. Multilingual Project. Version 5.0.2.1570*, StarUML, 2005. <http://staruml.sourceforge.net/en/> (consultado en abril de 2008)
- [9] Intalio, *Intalio Designer, herramienta de modelado con notación para modelado de procesos del negocio*. <http://www.intalio.com/> (consultado en mayo de 2008)
- [10] Stemberger, M., Jaklic, J. y Popovic, A., “Suitability of Process Maps for Business Process Simulation in Business Process Renovation Projects”, *16TH EUROPEAN SIMULATION Symposium ESS2004. Budapest (Hungary)*, pp. 109-123, 2004.
- [11] Eriksson, H. y Penker, M., *Business Modeling with UML. Business Patterns at Work*, Wiley, 2000.
- [12] Free Software Foundation, *The Free Software Definition*, 2007. <http://www.fsf.org/licensing/essays/free-sw.html> (consultado en enero de 2008)
- [13] Open Source Initiative, *The Open Source Definition*, 2007. <http://opensource.org/docs/osd> (consultado en enero de 2008)
- [14] FLOSS, *Free/Libre and Open Source Software: Survey and Study*. <http://www.flossproject.org/outline.htm> (consultado en enero de 2008)
- [15] QSOS Method, *QSOS Introduction*, 2006. http://www.qsos.org/?page_id=7 (consultado en abril de 2008)

- [16] List, E. y Korherr, B., “An Evaluation of Conceptual Business Process Modelling”, en: Haddad, H. M. (ed.), *Languages Symposium on Applied Computing. Proceedings of the 2006 ACM Symposium on Applied Computing*, pp. 1.532-1.539, 2006.
- [17] Pérez, M., Grimán, A., Mendoza, L. y Rojas, T., “A Systemic Methodological Framework for IS Research”, *Proceedings of the Tenth Americas Conference on Information Systems AMCIS. New York (USA)*, pp. 4.374-4.383, 2004.
- [18] Baskerville, R., “Investigating Information Systems with Action Research”, *Communications of the Association for Information Systems*, vol. 2, nº 19, pp. 1-32, 1999.
- [19] Kitchenham, B., “Evaluating Software Engineering Methods and Tools. Part 1: The Evaluation Context and Evaluation Methods”, *ACM Software Engineering Notes*, vol. 21, nº 1, pp. 11- 14, 1996.
- [20] Ortega, M., Pérez, M. y Rojas, T., “Construction of a Systemic Quality Model for Evaluating a Software Product”, *Software Quality Journal*, vol. 11, nº 3, pp. 219-242, 2003.
- [21] Pérez, M., Domínguez, K., Mendoza, L. y Grimán, A., “Human Perspective in System Development Quality”, *Proceedings of the Twelfth American Conference on Information Systems. AMCIS2006. Acapulco (México)*, pp. 3.823-3.834, 2006.
- [22] Kirchner, L. y Jung, J., “A Framework for the Evaluation of Meta-Modelling Tools”, *The Electronic Journal Information Systems Evaluation*, vol. 10, nº 1, pp. 65-72, 2007.

X Jornadas de Innovación y Calidad del Software Conferencia Iberoamericana de Calidad del Software

EOI Escuela de Negocios, Madrid, 24-25 de septiembre de 2008

Organizadas por el grupo de Calidad del Software de
ATI (www.ati.es/gtcalidadsoft) en colaboración con EOI



Patrocinadores Premium:



Medio colaborador:



Colaboran:



Con el apoyo (Orden ITC/390/2007) institucional de:

