

Revista
Española de
Innovación,
Calidad e
Ingeniería del Software



Volumen 7, No. 2, octubre, 2011

Web de la editorial: www.ati.es

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

E-mail: calidadsoft@ati.es

ISSN: 1885-4486

Copyright © ATI, 2011

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 (ATI), Via Laietana, 46, 08003 Barcelona.

Secretaría de dirección: ATI Madrid, C/Padilla 66, 3º dcha., 28006 Madrid



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

Editor

Dr. D. Luís Fernández Sanz (director)

Departamento de Ciencias de la Computación, Universidad de Alcalá

Miembros del Consejo Científico

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. Tanja Vos

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

Dña. M^a del Pilar Romay

CEU Madrid

Dr. D. Alvaro Rocha

Universidade Fernando Pessoa
Porto

Dr. D. Oscar Pastor

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

Dra. Dña. María Moreno

Depto. de Informática
Universidad de Salamanca

Dra. D. Javier Aroba

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

D. Guillermo Montoya

DEISER S.L.
Madrid

Dr. D. Pablo Javier Tuya

Depto. de Informática
Universidad de Oviedo

Dra. Dña. Antonia Mas

Depto. de Informática
Universitat de les Illes Balears

D. Jacques Lecomte

Meta 4, S.A.
Francia

Dra. Raquel Lacuesta

Depto. de Informática e Ing. de Sistemas
Universidad de Zaragoza

Dra. María José Escalona

Depto. de Lenguajes y Sist. Informáticos
Universidad de Sevilla

Dr. Dña. Aylin Febles

CALISOFT
Universidad de Ciencias Informáticas (Cuba)

Contenidos

REICIS

Editorial	4
<i>Luís Fernández-Sanz</i>	
Presentación	5
<i>Luis Fernández-Sanz</i>	
Evidencia empírica sobre mejoras en productividad y calidad en enfoques MDD: un mapeo sistemático	6
<i>Yulkeidi Martínez, Cristina Cachero y Santiago Meliá</i>	
Guía de pruebas de software para MoProSoft	28
<i>Silvia Guardati y Alain Ponce</i>	
Sección Actualidad Invitada:	48
Aseguramiento de calidad del software en Administraciones Públicas	
<i>Marcos Blanco, CESJE</i>	

Evidencia empírica sobre mejoras en productividad y calidad en enfoques MDD: un mapeo sistemático

Yulkeidi Martínez
Universidad de Ciego de Ávila, Cuba
yulkeidi@gmail.com
Cristina Cachero, Santiago Meliá
Universidad de Alicante, España.
{ccachero, santi}@dlsi.ua.es

Resumen

Para el avance del desarrollo de software dirigido por modelos, es esencial proporcionar evidencias empíricas que corroboren o refuten las promesas de mejora asociadas a este paradigma desde su concepción. El objetivo de este trabajo es clasificar la evidencia empírica existente respecto de la mejora en productividad y calidad de las aplicaciones. Para ello hemos aplicado el proceso de mapeo sistemático, un tipo de estudio secundario diseñado específicamente para abordar este tipo de objetivos. Como resultado de este trabajo, hemos identificado asunciones que carecen a día de hoy de evidencia empírica. Por tanto, constituyen líneas de trabajo que se deben abordar para una mayor rigurosidad y consistencia de la disciplina. También hemos identificado áreas donde un análisis más exhaustivo podría ser de utilidad. El mapa resultante facilita la entrada de nuevos investigadores a este campo.

Palabras clave: Mapeo sistemático, ingeniería dirigida por modelos, calidad, productividad

Empiric evidence on productivity and quality improvements with MDD approaches: a systematic mapping

Abstract

In order to consolidate the progress in the development of the Model-Driven Development paradigm, it is essential to provide empirical evidence that either corroborates or refutes the promises of improvement that attached to this paradigm since its inception. The purpose of this paper is to classify the existing empirical evidence referred to improvements in productivity and quality of the applications. In order to achieve this goal, we have applied the systematic mapping process, a type of secondary study specifically devoted to carry out this kind of studies. As a result of this work, we identified assumptions based on today's lack of empirical evidence, and therefore lines of work are to be addressed to bring more rigor and consistency in discipline. We have also identified areas where further analysis could be useful. The resulting map facilitates the entry of new researchers to the field.

Key words: Systematic Mapping, Model Driven Development, Quality, Productivity

Martínez, Y., Cachero, C. y Meliá, S., "idencia empírica sobre mejoras en productividad y calidad en enfoques MDD: un mapeo sistemático", REICIS, vol. 7, no.2, 2011, pp. 6-28. Recibido: 21-2-2011; revisado: 11-7-2011; aceptado: 15-9-2011.

1. Introducción

El Desarrollo Dirigido por Modelos (MDD, del inglés *Model Driven Development*) es una aproximación al desarrollo de software basado en (a) la creación de modelos del sistema software a distintos niveles de abstracción y (b) su uso como base de un proceso de generación automática de código [1]. Entre las reivindicaciones de este paradigma de desarrollo se encuentran [2] [3] [4]:

1. Mayor simplicidad del proceso. El desarrollador se puede aislar de la complejidad tecnológica y centrarse en la estructura y comportamiento deseado de la aplicación.
2. Mejora de la productividad del proceso de desarrollo. El uso de modelos independientes de cómputo (CIM, *Computation-Independent Model*), modelos independientes de plataforma (PIM, *Platform-Independent Model*) y modelos de plataforma específica (PSM, *Platform-Specific Model*) permiten especificar el sistema a distintos niveles de abstracción y favorecen de este modo el reuso. La definición de transformaciones modelo a modelo y modelo a código automatiza gran parte del proceso de codificación.
3. Mejora de la calidad externa de la aplicación resultante (Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad) [5].

Sin embargo, y a pesar de las numerosas llamadas de atención por parte de la comunidad de Ingeniería del Software acerca de la necesidad de acompañar este tipo de afirmaciones con evidencias empíricas [6] [7], la gran mayoría de aportaciones en el campo del MDD sigue tomando la forma de nuevas metodologías, técnicas y herramientas que, aunque viables, no llegan a demostrar de una manera fiable su utilidad y ventajas respecto a sus predecesoras.

Con el fin de ayudar a que la comunidad proporcione estas evidencias empíricas, el área de experimentación en Ingeniería del Software ha desarrollado guías exhaustivas que ayudan a los investigadores en el proceso de obtención de datos fiables acerca de las ventajas o desventajas de los distintos métodos, técnicas o herramientas empleadas en la construcción de sistemas software [8] [9]. En ausencia de estos datos, se sigue corriendo el peligro de sostener conclusiones erróneas [10], perjudicando de esta manera tanto la toma

de decisiones en el ámbito empresarial [11] como la propia imagen de la disciplina, tal y como ya ha sucedido en el pasado [12].

Aún más importante, es necesario proporcionar un acceso rápido, claro y conciso a las evidencias empíricas de las que se dispone, de manera que ese conocimiento llegue a los encargados de decidir acerca de su adopción en la práctica. En el caso de MDD, esta falta de organización de la evidencia empírica, así como, cuando existe, su falta de relación con metodologías bien definidas, puede estar perjudicando su adopción por parte de las empresas; es bien sabido que la decisión de adoptar una nueva aproximación o utilizar nuevas herramientas en un proceso de desarrollo de software debería venir avalada por un proceso fiable y repetible, de manera que se maximicen las probabilidades de éxito de la implantación en la industria [13].

Con el objetivo de (a) atraer la atención de la comunidad de Ingeniería del Software sobre la falta de un acervo empírico en el campo de MDD y (b) proporcionar un mapa conceptual que organice los datos que se han publicado hasta el momento, este trabajo presenta un mapeo sistemático [14] [15] de la evidencia empírica existente acerca de cómo MDD contribuye a mejorar la calidad de la aplicación resultante y la productividad del proceso de desarrollo. Esta evidencia se relaciona directamente con las reivindicaciones 2 y 3 presentadas al inicio de esta sección.

El trabajo está organizado como sigue: en la Sección 2 se presenta en detalles el proceso de del mapeo sistemático. En la sección 3 se presenta el análisis comparativo y se discuten los resultados del mapeo sistemático y sus limitaciones. Por último, en la sección 4 se presentan las conclusiones y trabajos futuros.

2. Mapeo Sistemático

La técnica de mapeo sistemático (*systematic mapping*) define un proceso y una estructura de informe que permite categorizar los resultados que han sido publicados hasta el momento en un área determinada [15]. El objetivo de un mapeo sistemático está en la clasificación, y está por tanto dirigido al análisis temático y a la identificación de los principales foros de publicación. Permite responder preguntas genéricas como ¿Qué es lo que se ha hecho hasta el momento en el campo X? Como limitación, este tipo de estudios no toma en consideración la calidad de los estudios incluidos. Una alternativa al mapeo

sistemático es la revisión sistemática [14], cuya fase de revisión de trabajos, mucho más rigurosa, permite establecer el estado de evidencia a través de la exhaustiva extracción de datos cuantitativos y estudios de meta-análisis, y por tanto responder a preguntas de investigación mucho más específicas. Estos dos tipos de estudios son complementarios y tienen como objetivo identificar los huecos de la investigación, por lo que un mapeo sistemático es considerado por muchos como un paso previo imprescindible para decidir en qué áreas concretas del campo es interesante abordar una revisión sistemática más detallada [16]. El proceso de mapeo sistemático seguido en la presente investigación se presenta en la figura 1.

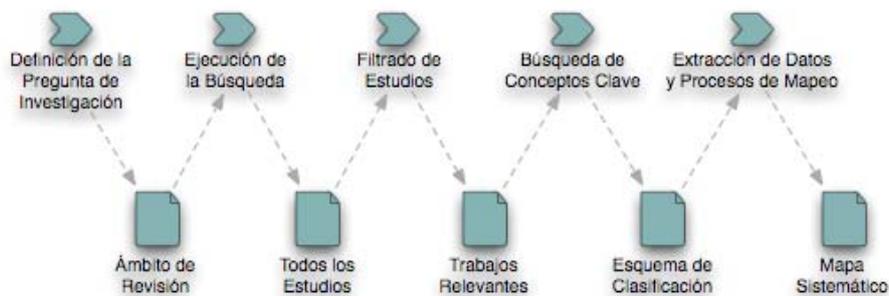


Figura 1. Proceso de mapeo sistemático.

2.1. Definición de la pregunta de investigación.

De acuerdo a [14], una pregunta de mapeo sistemático bien focalizada incluye cuatro partes:

- | | |
|---------------------------------|--|
| Población: | Profesionales interesados en migrar hacia procesos de desarrollo basados en el paradigma MDD. |
| Factor de estudio: | Impacto de MDD sobre la calidad externa del software y la productividad del proceso. |
| Intervención en la comparación: | Marcos de trabajo, metodologías, herramientas, etc. basadas en el paradigma MDD que han servido de base para comparar productividad del proceso y/o calidad externa de las aplicaciones resultantes con respecto a otros paradigmas. |
| Resultado: | Grado de evidencia empírica existente en el campo. |

Estos elementos nos han permitido definir las siguientes preguntas de investigación (PI):

- PI1: ¿Qué tipos de publicación ofrecen más datos empíricos sobre el impacto de MDD sobre la calidad y la productividad, y cómo ha cambiado la tendencia a lo largo del tiempo?
- PI2: ¿Qué características de calidad de producto (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad [5]) y proceso han sido más investigadas desde un punto de vista empírico?
- PI3: ¿Qué tipo de estudio empírico (encuestas, casos de estudio, experimentos, meta-análisis) es el más usual a la hora de aseverar el impacto de MDD sobre características de calidad y productividad?
- PI4: ¿Qué tipo de enfoque de investigación (validación en entornos controlados o evaluación en entornos reales) es el más utilizado en el campo?

En base a estas preguntas, el objetivo de la investigación presentada en este artículo se puede resumir como: “identificar los estudios empíricos que se han realizado durante el período 2000-2010 sobre la mejora de la calidad del producto y la productividad del proceso mediante el uso de aproximaciones MDD”.

2.2. Ejecución de la Búsqueda.

La cadena de búsqueda utilizada como base para la obtención de los trabajos relevantes ha sido: *empirical AND software AND (quality or performance) AND ("model driven" OR model-driven OR MDD OR MDE OR MDA) AND (experiment OR survey OR "case study" OR meta-analysis)*. La búsqueda se ha centrado en los años 2001-2010. La elección del período temporal está motivada por el hecho de que el 2001 fue el año en que la OMG (*Object Management Group*) propuso la adopción de MDA (*Model Driven Architecture*) como estándar para las actividades involucradas en el MDD, aunque la guía oficial no fue publicada hasta junio 2003 [17].

Por otro lado, las fuentes de datos utilizadas han sido cuatro: *Google Scholar*, ACM, IEEE y *Springer*. Esta selección de fuentes es hasta cierto punto redundante, ya que *Google Scholar*, motor de búsqueda líder en el seno de la comunidad científica de investigadores académicos, indexa un gran número de fuentes de documentación técnica, entre las que se encuentran ACM, IEEE y *Springer*, que a su vez son los tres foros más significativos en la Ingeniería del Software. *Google Scholar* recupera además documentos que no aparecen en bibliotecas digitales organizadas, y que sin embargo forman parte del acervo científico de

las distintas disciplinas, por lo que consideramos que es un complemento importante en la elaboración de mapeos y revisiones sistemáticas [18]. No menos importante, *Google Scholar* muestra los documentos ordenados en función de la importancia del foro de publicación en el que se encuentra el artículo, la relevancia de sus autores o la frecuencia con la que es citado el escrito. Por tanto su inclusión nos ayuda a garantizar que estamos incluyendo los trabajos más relevantes para la comunidad científica.

Los resultados de ejecutar nuestra cadena de búsqueda en *Google Scholar* fueron 896 artículos, que incluyen, como ya hemos comentado, tanto publicaciones indizadas en las librerías digitales más importantes en el área de Ingeniería del Software (incluidas ACM, IEEE y *Springer*) como literatura gris publicada on-line en las Webs de las universidades (informes técnicos, tesis de grado, maestrías, tesis doctorales, etc.). Una revisión preliminar de los resultados nos permitió constatar que las publicaciones relevantes para nuestro estudio se encontraban concentradas en los primeros puestos de la búsqueda, por lo que se decidió limitar el análisis más exhaustivo de las mismas a los 300 primeros trabajos.

Nuestro segundo paso de búsqueda consistió en adaptar la cadena de búsqueda a la idiosincrasia de los buscadores específicos de ACM, IEEE y *Springer*. La ejecución de la búsqueda en estos motores específicos arrojó 99 resultados (ACM), 3136 resultados (IEEE) y 317 resultados (*Springer*) respectivamente. Nuevamente, dado que los tres los buscadores ordenan por relevancia de la publicación, se analizaron los 100 resultados más relevantes de cada buscador, lo que nos da un total de 599 publicaciones analizadas. En la Tabla 1 se presenta un resumen de estos resultados.

Buscador	<i>Google Scholar</i>	ACM	IEEE	<i>Springer</i>	Total
Resultados de la Consulta	896	99	3136	317	4448
Trabajos analizados	300	99	100	100	599
Trabajos candidatos	187	76	52	39	354
Trabajos Relevantes	40	12	6	6	64
Coincidencias con <i>Google Scholar</i>	-	9	1	4	14
Total Trabajos Relevantes	40	3	5	2	50

Tabla 1. Resultados de la búsqueda antes y después de eliminar duplicados.

2.3. Filtrado de Estudios

El protocolo inicial de revisión definido para la selección de los estudios primarios se ha formulado basado en los siguientes criterios de inclusión/exclusión:

Inclusión: libros, documentos, informes técnicos y la literatura gris que describe los estudios empíricos sobre calidad o productividad en MDD, incluso aunque solo se tenga acceso al resumen del mismo.

Exclusión:

1. Artículos que no reportan estudios empíricos acerca de las mejoras en calidad o productividad cuando se utiliza una aproximación MDD. Esto implica dejar de lado cualquier trabajo centrado en justificar la mera viabilidad de la propuesta (sin comprobación empírica de las mejoras que introducen en cuanto a productividad y calidad externa del producto final).
2. Discusiones teóricas, revisiones y clasificaciones, así como propuestas de modelos de calidad que no vienen acompañados de un estudio empírico.
3. Estudios que se centran en evaluar/mejorar la calidad de los modelos y/o transformaciones que intervienen en las aproximaciones MDD, salvo que lo hagan en relación a su impacto sobre la calidad del producto final.
4. Estudios empíricos sobre aproximaciones MDD que estudian factores de contexto o datos subjetivos (e.g. los que evalúan los factores que influyen en su adopción exitosa, opinión subjetiva de desarrolladores, etc.).

2.3.1 Fiabilidad del criterio de inclusión

Para evaluar la fiabilidad de los criterios de inclusión/exclusión de los estudios relevantes, y por tanto incrementar las posibilidades de obtener resultados fiables e independientes del evaluador, se seleccionó una submuestra de la población, consistente en las 100 primeras referencias arrojadas por la búsqueda inicial en *Google Scholar*, lo que supone un 11,16% de la población total de los resultados arrojados por este buscador, y un 33% de los resultados finalmente analizados del mismo. Tras el establecimiento de los criterios de revisión, el título y el resumen de dichas referencias fueron utilizados para clasificar los trabajos de manera independiente por dos revisores: E1 y E2. La fiabilidad inter-evaluador se calculó mediante el estadístico Kappa de *Cohen* [19]. El grado de fiabilidad arrojado por el estadístico fue satisfactorio (Kappa=0,811, ver tabla 2). Este grado de acuerdo indica la existencia de una base de criterios suficientemente clara y que no denota divergencias significativas entre los revisores [20].

		E1		Total
		0	1	
E2	0	86	2	88
	1	2	10	12

Tabla 2. Tabla de Contingencia. 2^{da} fase del proceso de inclusión/exclusión (título + resumen)

Durante la fase de conciliación de diferencias entre los evaluadores resultó especialmente interesante constatar la poca homogeneidad en cuanto al formato de informes de resultados empíricos, así como el uso tan libre que se sigue haciendo de palabras como “caso de estudio” o “experimento” en la literatura de Ingeniería del Software, a pesar de las numerosas llamadas de alerta al respecto [20]. En la práctica, esas palabras se refieren en muchas ocasiones a meros estudios de viabilidad de la propuesta, lo que complica notablemente la obtención de resultados relevantes en una búsqueda como la planteada en el presente artículo.

Los criterios validados fueron aplicados a los resultados devueltos por los cuatro motores de búsqueda utilizados como fuentes de datos para el mapeo sistemático. Finalmente se incluyeron en el estudio un total de 50 trabajos: 40 de ellos aparecían en los resultados de *Google Scholar*, 3 en *ACM*, 5 en *IEEE* y 2 en *Springer*. En la tabla 1 se presentan los datos pormenorizados. Analizando dicha tabla se aprecia que el grado de coincidencia de resultados entre el buscador genérico y los buscadores específicos apenas alcanza un 55%, lo que demuestra la complementariedad de *Google Scholar* y el resto de motores de búsqueda [18]. Además, es interesante notar cómo *IEEE* aparece como la fuente de información peor cubierta por *Google Scholar* (ver tabla 1).

2.4. Definición del esquema de clasificación.

Una vez seleccionados los trabajos relevantes se definieron, en base a los objetivos del estudio, tres tipos de clasificaciones (ver figura 2):

- Medidas evaluadas: de calidad de proceso (productividad) y de calidad de producto (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad). Esta última clasificación se basa en el modelo de calidad presentado en la norma ISO/IEEE 9126 [5].
- Tipo de estudio empírico realizado [21] [22]: caso de estudio, experimento, encuesta o meta-análisis.

- Enfoques de investigación: validación en entornos controlados o evaluación en entornos reales [23].

Además, para cada estudio relevante se recopiló información referente al año de publicación y tipo de publicación mediante el que fue diseminado.

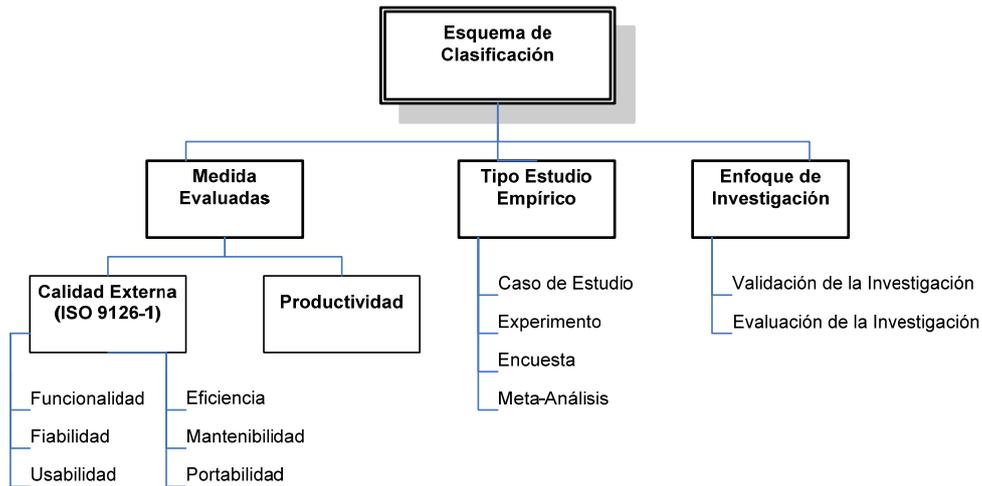


Figura 2. Esquema de clasificación.

2.5. Extracción de datos y Mapeo sistemático.

Tras definir el sistema de clasificación, el último paso del mapeo sistemático consiste en la extracción de datos y el proceso de mapeo de las distintas dimensiones. El resultado completo de esta actividad se muestra en el Apéndice A. El resultado sintetizado de nuestro estudio se puede observar de manera gráfica en el diagrama de burbuja de la figura 3, que visualiza: (1) la relación entre el tipo de estudio empírico y el estudio de medidas de calidad impactadas por el uso del paradigma MDD y (2) la frecuencia de estudios empíricos por años de publicación.

La figura 3 ilustra básicamente dos diagramas de dispersión XY con burbujas en las intersecciones de categoría, que permite tener en cuenta varias categorías al mismo tiempo y da una visión general rápida de un campo de estudio, proporcionando un mapa visual [15]. En esta visualización de los resultados, el tamaño de una burbuja es proporcional al número de artículos que están en el par de categorías que correspondan a la burbuja de las coordenadas. Cuando un trabajo ha afectado más de una categoría (e.g. un meta-análisis de más de una característica de calidad, o presentación de más de un tipo de estudio empírico),

el trabajo ha sido contabilizado en todas las características. De este modo, los porcentajes dan una visión más real sobre la cantidad de esfuerzo invertido en cada una de las dimensiones de calidad incluidas en el estudio.

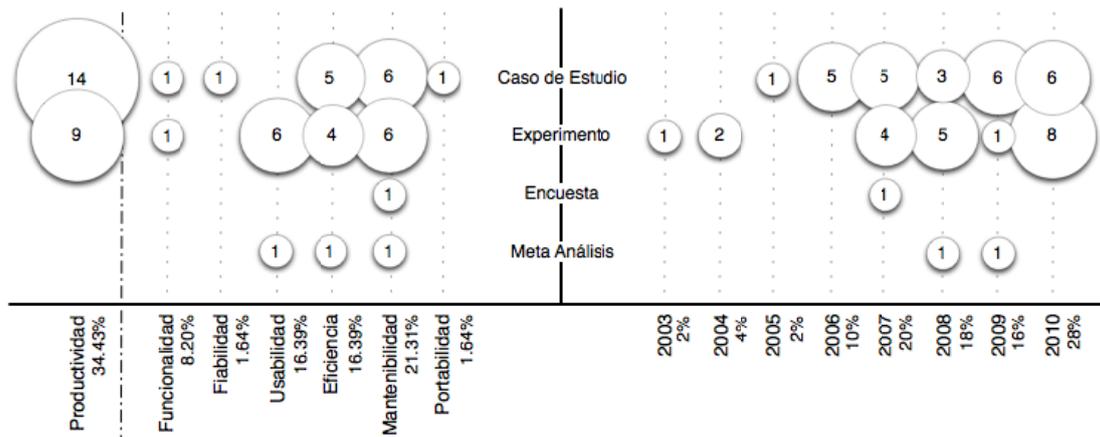


Figura 3. Diagrama de burbujas. Visualización del mapeo sistemático.

De igual forma, en la figura 4 se puede observar la distribución de trabajos por tipo de publicación y por enfoque de investigación. Del total de publicaciones incluidas en el mapeo (50), 21 pertenecen a revistas, 20 fueron presentadas en conferencias y 9 provienen de otros tipos de publicaciones como libros, tesis o informes técnicos. Por otro lado, 28 consistieron en validaciones en ambiente controlado, frente a 22 evaluaciones (en entornos de uso real) del paradigma.

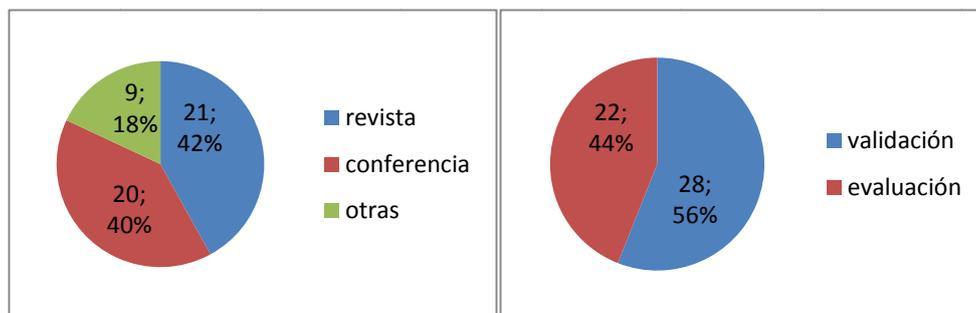


Figura 4. Diagramas Circulares. Artículos por tipo de publicación y por enfoque de investigación.

3. Análisis comparativo y discusión

A continuación damos respuesta a las preguntas de investigación formuladas en la Sección 2.1. a través de los resultados obtenidos.

- P11: Tras un análisis pormenorizado de los resultados obtenidos, se puede apreciar que la publicación de estudios empíricos en el ámbito de MDD presenta una

tendencia al alza, en consonancia con lo que está ocurriendo en otros campos de la Ingeniería del Software. Un indicador de esta tendencia es que aproximadamente 28% de los estudios empíricos realizados en MDD se concentran en el año 2010. Los foros preferidos para este tipo de trabajos son revistas (42%), lo que indica la relevancia otorgada por los investigadores a los resultados obtenidos. Otro dato interesante arrojado por el estudio es que, de las 599 publicaciones candidatas a ser incluidas en el estudio, solo 50 (un 8,3%) cumplieron los criterios para ser finalmente incluidas en el análisis, lo que, dado la especificidad de la búsqueda, por un lado demuestra el uso tan libre que se hace de los términos empíricos en el área, y por otro es un indicador de la falta de madurez empírica de MDD, lo que nos lleva a pensar que puede ser pronto todavía para embarcarse en una revisión sistemática de dicha evidencia empírica. En caso de realizarse, los campos más prometedores serían el impacto de MDD en productividad y mantenibilidad.

- PI2: La mantenibilidad (ver figura 3) con un 21,31% de los resultados centrados en ella, es la característica de calidad externa que más ha sido investigada en el enfoque MDD. No lejos se encuentran la eficiencia y la usabilidad, con un 16,39% cada una. El resto de las medidas han sido estudiadas en menor escala, lo que plantea una oportunidad para futuras tesis y trabajos de investigación. Por otro lado, el estudio empírico de la productividad, como medida de calidad del proceso de desarrollo (eficiencia de proceso) ha tenido un auge representativo en los últimos 3 años, con un 34,43% de los resultados empíricos centrados en ella. Este dato parece sugerir que el trabajo empírico en MDD se orienta a demostrar la calidad de proceso más que la calidad de producto, lo que es consistente con la preponderancia de reivindicaciones de mejora del proceso entre las razones esgrimidas por los investigadores para la adopción del paradigma.
- PI3: Como se puede observar en la figura 3, los casos de estudio y los experimentos resultan ser los estudios empíricos con mayor predominio en las investigaciones actuales. Este es un dato prometedor, ya que estos estudios, donde se mide lo que hacen los sujetos, en lugar de medir lo que dicen que hacen, ofrecen un mayor grado de fiabilidad. Además, es interesante constatar cómo la usabilidad ha sido estudiada

exclusivamente mediante experimentos, cuyos resultados son los más fiables a costa de un alcance necesariamente más limitado.

- PI4: En cuanto a los enfoques de investigación más utilizados, los números (ver figura 4) indican que la mayoría de las investigaciones actuales sobre el uso del enfoque MDD han demostrado la aplicabilidad de su propuesta a través de validaciones (estudios en entornos sintéticos). En concreto, 28 publicaciones de las 50 (56%) han utilizado este enfoque, lo que aumenta la fiabilidad de los resultados pero no permite identificar con exactitud los beneficios e inconvenientes de su aplicación en entornos reales de desarrollo (al contrario de lo que ocurre con la evaluación de la investigación) [23]. Además, durante el último año incluido en el estudio (2010) se ha incrementado la cantidad de estudios evaluados empíricamente tanto en entornos sintéticos como en la industria (ver Apéndice A, tabla A.1), lo que nos permite afirmar que la comunidad investigadora está siendo cada vez más cuidadosa en propiciar investigaciones claras, reproducibles y con garantías de aplicación industrial.

3.1 Limitaciones del estudio

La principal limitación de este mapeo sistemático es haber analizado sólo 599 publicaciones del total de 4448 resultados obtenidos a las consultas realizadas en los diferentes buscadores (ver tabla 1). Aunque esta decisión sin duda perjudica la cobertura del estudio, creemos que esta desventaja se ve paliada por el uso de los buscadores más relevantes del campo (*Google Scholar*, ACM, IEEE y *Springer*) y por la ordenación que hacen dichos buscadores de los resultados en función de su relevancia para la búsqueda.

La segunda limitación, debida a la propia filosofía del mapeo sistemático, es la calidad de los estudios incorporados. Esta limitación se podría haber evitado realizando una revisión sistemática de dichos trabajos. Sin embargo, el bajo número de publicaciones verdaderamente pertinentes (50), nos hace pensar que es todavía pronto para este tipo de evaluaciones de calidad.

Otra posible limitación de este tipo de estudios es la posibilidad de errar la clasificación por el uso ambiguo que hacen los autores de conceptos como, en nuestro caso, experimento o caso de estudio. Relacionado con esto, hemos ratificado lo que ya advertían otros autores [15] acerca de los resúmenes de los artículos, que a menudo son engañosos y

carecen de información importante. En este estudio se ha limitado el impacto de estos dos riesgos mediante una aproximación conservadora al proceso de inclusión/exclusión de estudios, que ha implicado la lectura de cuantas partes del artículo hayan sido necesarias hasta poder resolver la duda de si incluir o excluir el estudio.

4. Conclusiones y trabajos futuros

Es un hecho que los estudios empíricos en el área del MDD se han incrementado sustancialmente en los últimos años, lo cual permite ya contar con ciertas evidencias del impacto de este paradigma sobre la calidad y productividad de los productos de software, más allá de resultados anecdóticos.

Sin embargo, el porcentaje de estudios que realmente proporcionan evidencia empírica acerca de mejoras de calidad y productividad en MDD sigue siendo muy bajo (solo la productividad del paradigma ofrece un número de estudios comparativamente significativo), lo que contrasta con otras disciplinas e incluso con otras áreas de Ingeniería del Software [24]. Por tanto, sigue siendo relevante enfatizar la necesidad de que la comunidad de MDD dedique una cantidad sustancial de esfuerzo a la comprobación empírica de sus aseveraciones que permita la posterior realización de revisiones sistemáticas y meta-análisis rigurosos.

De manera general, los resultados más relevantes reportados por los estudios empíricos relacionados con la productividad de MDD se pueden resumir como sigue:

- Los estudios que validan el uso de MDD en entornos académicos reportan una productividad de 2 a 9 veces superior a la obtenida con otros paradigmas de desarrollo [25]. La productividad puede llegar a ser hasta 20 veces superior según se va aumentando el tamaño del proyecto de desarrollo. Estos resultados contrastan con los reportados por experimentos en entornos industriales [26], donde los resultados son mucho más heterogéneos, y van desde los que directamente reportan una pérdida de productividad de un 10% [27] a los que coinciden con los estudios académicos y reportan ganancias que oscilan entre un 20 % y un 35 % de productividad [27] [28] [29] [30]. Como principal limitación a estos resultados, los informes en entornos industriales se basan en estudios a pequeña escala. Las principales razones esgrimidas para justificar las pérdidas de productividad se

relacionaban directamente con el uso de herramientas inmaduras y altos costos de modelado, que puede llegar a ser tan complejo como programar directamente con un lenguaje tradicional de tercera generación.

- Por otro lado, se ha constatado cómo el uso de herramientas, librerías especializadas, etc. de soporte a MDD reduce la curva de aprendizaje y el esfuerzo de entrenamiento con este paradigma, lo que a su vez incide en la productividad del paradigma [31].
- Existen evidencias de que el enfoque MDD presenta deficiencias a la hora de expresar las reglas de diseño arquitectónicas de la aplicación, lo que actúa en detrimento de la productividad y la calidad de la aplicación final [32].

Con respecto a la mantenibilidad, los artículos incluidos en este estudio reportan que el tiempo necesario para evaluar el impacto de un cambio es substancialmente más corto (con un decremento que, en los estudios revisados, ronda el 37%) si se usa una visualización gráfica (base del MDD) en comparación con una textual (no MDD) [33]. Además, usando aproximaciones MDD completas para el proceso de mantenimiento se mejora aún más la mantenibilidad [34]. Estos datos contrastan en cierta medida con lo reportado en [35], donde se expone que el uso de UML como lenguaje de modelado para tareas de mantenimiento no tiene un impacto significativo en el tiempo necesario para realizar un cambio, pues también se debe contar el tiempo para poner al día la documentación de UML. Sin embargo, por lo que se refiere a la exactitud funcional de los cambios (introducción de errores en el software durante el cambio), UML ha demostrado un impacto positivo a la hora de mejorar la calidad del código, incluso si los desarrolladores no están muy familiarizados con su uso. Otros estudios [36] [37] [38] presentan experimentos para medir el efecto del reuso, la complejidad y el acoplamiento de objetos en la mantenibilidad del software, aunque sin comparar distintas aproximaciones:

- La reusabilidad del software aumenta en detrimento de la simplicidad y mantenibilidad de algunos artefactos [36].
- Existe correlación entre la complejidad de un diagrama y las facilidades de comprensión y modificación del mismo [37].

- De igual forma, el acoplamiento entre objetos está fuertemente correlacionado con las facilidades de comprensión y modificación de expresiones OCL que pueden actuar como aserciones en el código final [38].

Si nos centramos en dominios concretos, una comparación del Modelado Orientado a Aspectos (MOA) vs la Codificación Orientada a Aspectos, los estudios experimentales y cuantitativos realizados en [39] reportan que las aproximaciones MOA generan una aplicación más pequeña, menos compleja y más modular. Además, en algunos casos el desarrollo basado en MOA acorta el ciclo de mantenimiento.

En cuanto al impacto de MDD sobre la eficiencia, funcionalidad, fiabilidad, usabilidad y portabilidad, la literatura provee resultados en su mayor parte anecdóticos, poco contrastados, lo que hace muy difícil su generalización. Los estudios se centran principalmente en presentar experiencias donde prácticas específicas (una determinada herramienta, aproximación, técnica, modelo, etc.) han mostrado sus bondades a la hora de mejorar la calidad del producto final.

A pesar de que nuestro estudio abarca el período comprendido entre los años 2001 y 2010, el trabajo empírico en el campo sigue dando frutos, y en los últimos meses se han seguido publicando estudios empíricos relevantes para nuestra investigación. En concreto, en [40] [41] se ha estudiado el impacto de los factores sociales, técnicos y orgánicos en el éxito o fracaso de la aproximación MDD en la industria. Particularmente se han encontrado una correlación positiva entre un entorno con mejores conocimientos organizativos y mejor comunicación en el equipo de desarrollo y las variables de productividad y mantenibilidad del proceso MDD, con un incremento de 66,7% y 73,7% respectivamente. También se han recopilado datos que parecen sugerir que el uso de modelos en la comprensión de problemas con alto nivel de abstracción incide en el aumento tanto de la productividad (hasta un 72,2%) como en la mantenibilidad (hasta un 73,4%). Otras conclusiones de estos estudios inciden en la importancia del entrenamiento y la educación para alcanzar las promesas de este paradigma.

Con el mapeo sistemático presentado en este trabajo se han conseguido identificar los principales trabajos de investigación empírica publicados hasta el momento en los principales foros científicos. Este tipo de estudios es básico de cara a facilitar la apertura

del campo a nuevos investigadores. Además, este estudio ha puesto de manifiesto las enormes posibilidades de trabajo en este campo para la comunidad de Ingeniería del Software empírica con respecto al impacto del uso de aproximaciones basadas en el paradigma MDD sobre la calidad tanto de producto como de proceso.

De especial relevancia resulta constatar los resultados tan dispares encontrados en función de si los estudios han sido realizados en un entorno académico o industrial y, sobre todo, en función del tamaño del proyecto abordado. Otras variables que, hasta lo que alcanza nuestro conocimiento, no han sido suficientemente consideradas y que podrían influir en los resultados son los tipos de aplicaciones abordadas (e.g. si se tratan de aplicaciones intensivas en datos, transaccionales, etc.) o la experiencia de los desarrolladores. Por último, sorprende no encontrar ningún meta-análisis de productividad del paradigma, pese a que sí existen meta-análisis de otras variables con menos contribuciones. Pensamos que esto es debido, por un lado, a la disparidad en cuanto a la calidad de los informes publicados, que a veces presentan carencias que hacen imposible evaluar la calidad del estudio y agregar los datos, y por otro a la heterogeneidad de los contextos y premisas sobre las que se han construido los estudios, lo que igualmente complica su comparación y la agregación de sus resultados. El uso de plantillas y paquetes experimentales estandarizados ayudaría sin duda a paliar estos problemas y a agilizar la obtención de datos de calidad.

Agradecimientos

A Jesús Pardillo, por sus útiles comentarios durante la elaboración de este trabajo.

Referencias

- [1] Muñoz, J. y Pelechano, V., “MDA vs Factorías de Software”. En: Estévez, A., Pelechano V. y Vallecillo, A. (eds.), *Actas del II Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM 2005)*. Granada (Spain), 13 de Septiembre, pp. 1, 2005.
- [2] Macías, M., “Modelamiento basado en el Dominio: Estado del Arte”, *Revista de las I Jornadas de Ingeniería de Software 2004*, vol. 1, nº 1, pp. 33-41, 2004.

- [3] Selic, B., “Model-Driven Development: Its Essence and Opportunities”. En: IEEE Computer Society Press (eds.), *Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'06)*. Gyeongju (Korea), April 24-26, pp. 313-319, 2006.
- [4] López, E., González, M., López, M. y Iduñate, E.L., “Proceso de Desarrollo de Software Mediante Herramientas MDA”, *Revista Iberoamericana de Sistemas, Cibernética e Informática*, vol. 3, nº 2, pp. 6-10, 2006.
- [5] ISO/IEC 9126-1, *Software engineering - Product quality Part 1: Quality model*, ISO, 2001.
- [6] Glass, R., “The Realities of Software Technology Payoffs”, *Communications of the ACM*, vol. 42, nº 2, pp. 74-79, 1999.
- [7] Pickard, M., *Combining Empirical Results in Software Engineering*. Technical Report vol. 1, Keele University, England, 2004.
- [8] Kitchenham, B., Linkman, S. y Law, D., “DESMET: a methodology for evaluating software engineering methods and tools”, *Computing and Control Engineering Journal*, vol. 8, nº 3, pp. 120-126, 1997.
- [9] Guerini, M. L., *Revisión de resultados experimentales en técnicas de prueba y de educación de conocimientos*. Tesis de Magister en Ingeniería de Software, Universidad Politécnica de Madrid, España, 2007.
- [10] Juristo N. y Moreno A., *Basics of Software Engineering Experimentation*, Kluwer Academic Publisher, 2001.
- [11] Picó, R., *Análisis funcional de la gestión de almacén y producción en openbravo ERP y su aplicación docente*. Proyecto fin de Carrera, Escuela Técnica Superior de Informática Aplicada, Universidad Politécnica de Valencia, España, 2010.
- [12] Vanderburg, G., Real Software Engineering, video/mp4 presentations of Lone Star Ruby Conference 2010, 2010. Disponible en: <http://confreaks.net/videos/282-lsrc2010-real-software-engineering>) [Consulta 17/10/2010].
- [13] Kriter Software SL., Los diez factores que garantizan el éxito de la implantación ERP, 2007. Disponible en: <http://www.kriter.net/item/es/software-empresa-erp-kriter-identifican-los-diez-factores-que-garantizan-el-exito-de-la-implantacion/18/25/> [Consulta 30/09/2010].

- [14] Kitchenham, B. y Charters, S., *Guidelines for performing systematic literature reviews in software engineering*. Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University, Inglaterra, 2007.
- [15] Petersen, K., Feldt, R., Mujtaba, S. y Mattsson, M., “Systematic mapping studies in software engineering”. En: Visaggio, G., Baldassarre, M.T., Linkman, S. y Turner, M. BCS eWIC (eds.), *Proceedings of 12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. University of Bari (Italy), June 26 - 27, pp. 71-80, 2008.
- [16] Mujtaba, S., Petersen, K., Feldt, R. y Mattsson, M., “Software product line variability: A systematic mapping study”, School of Engineering, Blekinge Inst. of Technology, 2008. Disponible en: sites.google.com/site/drfeldt/mujtaba_apsec08_sysmap_spl_variabili.pdf [Consulta 14/10/2010].
- [17] Object Management Group, *Overview and guide to OMG's architecture (MDA Guide v1.0.1)* OMG, 2003. Disponible en: www.omg.org/cgi-bin/doc?omg/03-06-01 [Consulta 15/12/2009].
- [18] Noruzi, A., “Google Scholar: The new generation of citation indexes”, *Libri/Copenhagen Journal*, vol. 55, nº 4, pp. 170-180, 2005.
- [19] Gwet, K., “Inter-Rater Reliability: Dependency on Trait Prevalence and Marginal Homogeneity”, *Series of Statistical Methods for Inter-Rater Reliability Assessment*, vol. 2, nº 2, pp. 1-9, 2002.
- [20] Fleiss, J.L., *Statistical methods for rates and proportions*, John Wiley, 1981.
- [21] Kish, L., “Some statistical problems in research design”, *American Sociological Review*, vol. 24, nº 3, pp. 328-338, 1959.
- [22] Mendes, E. y Mosley, N., *Web Engineering*, Springer-Verlag New York Inc, 2006.
- [23] Wieringa, R., Maiden, N., Mead, N. y Rolland, C., “Requirements engineering paper classification and evaluation criteria: a proposal and a discussion”, *Journal Requirements Engineering*, vol. 11, nº 1, pp. 102-107, 2006.
- [24] Zelkowitz, M., “An update to experimental models for validating computer technology”, *The Journal of Systems and Software*, vol. 82, nº 3, pp. 373–376, 2009.
- [25] Diaz, O. Villoria F.M., “Generating blogs out of product catalogues: An MDE approach”, *The Journal of Systems and Software*, vol. 83, nº 10, pp. 1970-1982, 2010.

- [26] Mohagheghi, P. y Dehlen, V., “Where is the proof? - A review of experiences from applying MDE in industry”. En: Schieferdecker, I. y Hartman, A. (eds.), *Proceedings of Model Driven Architecture: Foundations and Applications: 4th European Conference, (ECMDA-FA 2008)*, Berlin (Germany), June 09-12, pp. 432–443, 2008.
- [27] MODELWARE D5.3-1 Industrial ROI, Assessment and Feedback-Master Document. Revision 2.2, 2006. Disponible en: www.modelware-ist.org [Consulta 28/09/2010].
- [28] The Middleware Company, Model Driven Development for J2EE Utilizing a Model Driven Architecture (MDA) Approach: Productivity Analysis (White Paper), Report by the Middleware Company on behalf of Compuware, 2003. Disponible en: www.omg.org/mda/mda_files/MDA_Comparison-TMC_final.pdf [Consulta 28/09/2010].
- [29] MacDonald, A., Russell, D. y Atchison, B., “Model-Driven Development within a Legacy System: an Industry Experience Report”. En: IEEE Computer Society Press (eds.), *Proceedings of Australian Software Engineering Conference (ASWEC 2005)*. Brisbane (Australia), 29 March-1 April, 2005, pp. 14–22, 2005.
- [30] Lange, C. y Chaudron, M., "Interactive Views to Improve the Comprehension of UML Models - An Experimental Validation". En: *IEEE Computer Society Press* (eds.), *Proceedings of the 15th IEEE International Conference on Program Comprehension (ICPC '07)*. Banff, Alberta (Canada), June 26-29, pp. 221-230, 2007.
- [31] Zhu, L., Gorton I., Liu Y. y Bao Bui N., “Model driven benchmark generation for web services”. En: Di Nitto, E., Hall, R.J., Han, J., Han, Y., Polini, A., Sandkuhl, K. y Zisman A. (eds.), *Proceedings of the 2006 International Workshop on Service-Oriented Software Engineering (SOSE 2006)*. Shanghai (China), May 27-28, pp. 33-39, 2006.
- [32] Mattsson, A., Lundell, B., Lings, B y Fitzgerald, B., “Linking Model-Driven Development and Software Architecture: A Case Study”, *IEEE Transactions on Software Engineering*, vol. 35 n° 1, pp. 83-93, 2009.
- [33] Mellegård, N. y Staron, M., “Improving Efficiency of Change Impact Assessment Using Graphical Requirement Specifications: An Experiment”. En: Babar, M.A., Vierimaa, M. y Oivo, M. (eds.), *Proceedings of Product-Focused Software Process Improvement 11th International Conference (PROFES 2010)*. Limerick (Ireland), June 21-23, pp. 336–350, 2010.

- [34] Goldschmidt, T., Reussner, R. y Winzen, J., “A Case Study Evaluation of Maintainability and Performance of Persistency Techniques”. En: Schäfer, W., Dwyer, M.B. y Gruhn, V. (eds.), *Proceedings of the 30th International Conference on Software Engineering (ICSE 2008). Leipzig (Germany), May 10-18*, pp. 401-410, 2008.
- [35] Dzidek W.J., Arisholm, E. y Briand, L.C.A, “Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance”, *IEEE Transactions on Software Engineering*, vol. 34, nº 3, pp. 407-432, 2008.
- [36] Lucrédio, D., “Evaluating the impact of MDD in software reuse”, 2009, Disponible en: www.les.inf.puc-rio.br/ [Consulta 03/11/2010].
- [37] Manso, M. E., Cruz-Lemus, J. A. Genero, M. y Piattini, M., “Empirical Validation of Measures for UML Class Diagrams: A Meta-Analysis Study”. En: Chaudron, M.R.V. (ed.), *Proceedings of the Workshop in Model Software Engineering: Workshops and Symposia in MODELS 2008). Toulouse (France), September 28-October 3*, pp. 303-313, 2008.
- [38] Reynoso, L., Genero, M. and Manso, E., “Measuring Object Coupling in OCL Expressions: A Cognitive Theory-Based Approach”. En: Daponte, P. (ed.), *Proceedings of the 23rd IEEE Instrumentation and Measurement Technology Conference (IMTC 2006). Sorrento (Italy), April 24-27*, pp. 1087–1092, 2007.
- [39] Hovsepyan, A., Scandariato, R. Van Baelen, S., Berbers, Y. y Joosen, W., “From Aspect-Oriented Models to Aspect-Oriented Code? The Maintenance Perspective”. En: Jézéquel, J. M. y Südholt, M. (eds.), *Proceedings of the 9th International Conference on Aspect-Oriented Software Development (AOSD 2010). Rennes and Saint Malo (France), March 15 -19*, pp. 85-96, 2010.
- [40] Hutchinson, J., Rouncefield, M. y Whittle, J., “Model-driven engineering practices in industry”. En: Taylor, R.N., Gall, H. y Medvidovic, N. (eds.), *Proceeding of the 33rd International Conference on Software Engineering (ICSE 2011). Waikiki, Honolulu (Hawaii), May 21-28*, pp. 633-642, 2011.
- [41] Hutchinson, J., Whittle, J., Rouncefield, M. y Kristoffersen, S., “Empirical assessment of MDE in industry”. En: Taylor, R.N., Gall, H. y Medvidovic, N. (eds.), *Proceeding of the 33rd International Conference on Software Engineering (ICSE 2011). Waikiki, Honolulu (Hawaii), May 21-28*, pp. 471-480, 2011.

Apéndice A: Publicaciones relevantes clasificadas

Publicación	Año	Fuente	Tipo de estudio	Medida de Calidad	Enfoque de Investig.
The value of conceptual modeling in database development: An experimental investigation	2003	GS	E	Productividad	V
The Impact of Software Reuse and Incremental Development on the Quality of Large Systems	2004	GS	E	Productividad, Mantenibilidad	Ev
Configuring real-time aspects in component middleware	2004	GS	E	Eficiencia	Ev
Alfresco Content Display	2005	GS	CE	Productividad	Ev
From code centric to model centric software engineering: practical case study of MDD infusion in a systems integration company	2006	GS	CE	Productividad	Ev
Applying model-driven development to distributed real-time and embedded avionics systems	2006	GS	CE	Productividad	V
Analysis of crosscutting in model transformations	2006	GS	CE	Mantenibilidad	V
Applying system execution modeling tools to evaluate enterprise distributed real-time and embedded system QoS	2006	GS	CE	Eficiencia	V
Model driven benchmark generation for web services	2006	ACM	CE	Productividad	Ev
A model-driven architecture approach using explicit stakeholder quality requirement models for building dependable information systems	2007	GS	CE	Fiabilidad	V
Quicker: A model-driven QoS mapping tool for QoS-enabled component middleware	2007	GS	CE	Funcionalidad	V
An activity-based quality model for maintainability	2007	GS	CE	Mantenibilidad	V
A Case Study on Model-Driven and Conventional Software Development: The Palladio Editor	2007	GS	CE	Productividad	Ev
mTurnpike: a Model-driven Framework for Domain Specific Software Development	2007	GS	E	Eficiencia	V
Building measure-based prediction models for UML class diagram maintainability	2007	GS	E	Mantenibilidad	V
Measuring Object Coupling in OCL Expressions: A Cognitive Theory-Based Approach	2007	GS	E	Mantenibilidad	V
Reliable Effects Screening: A Distributed Continuous Quality Assurance Process for Monitoring Performance Degradation in Evolving Software Systems	2007	GS	CE	Eficiencia	Ev
Interactive views to improve the comprehension of UML models-an experimental validation	2007	IEEE	E	Productividad	V
Survey of traceability approaches in model-driven engineering	2007	IEEE	S	Mantenibilidad	Ev
Where is the proof?-a review of experiences from applying MDE in industry	2008	GS	CE	Productividad	V
Transitioning from code-centric to model-driven industrial projects-empirical studies in industry and academia	2008	GS	E	Productividad	V
Factors affecting developers' use of MDSD in the Healthcare Domain: Evaluation from the MPOWER Project	2008	GS	E	Funcionalidad, Usabilidad, Productividad	Ev
Usability evaluation of user interfaces generated with a model-driven architecture tool	2008	GS	E	Usabilidad	V
Analyzing the Influence of Certain Factors on the Acceptance of a Model-based Measurement Procedure in Practice: An Empirical Study	2008	GS	E	Usabilidad	V
A realistic empirical evaluation of the costs and benefits of uml in software maintenance	2008	IEEE	E	Mantenibilidad	Ev

A case study evaluation of maintainability and performance of persistency techniques	2008	IEEE	CE	Productividad Mantenibilidad	Ev
A Model-Based Framework for Security Policy Specification, Deployment and Testing	2008	Springer	CE	Eficiencia	Ev
Model-Driven Performance Evaluation for Service Engineering	2008	Springer	MA	Eficiencia	Ev
Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process	2009	GS	CE	Productividad	V
Embedded System Construction–Evaluation of Model-Driven and Component-Based Development Approaches	2009	GS	CE	Mantenibilidad	Ev
Evaluating the Correctness and Effectiveness of a Middleware QoS Configuration Process in Distributed Real-time and Embedded Systems	2009	GS	CE	Portabilidad	Ev
Level of detail in UML models and its impact on model comprehension: A controlled experiment	2009	GS	E	Usabilidad	V
Empirical Validation of Measures for UML Class Diagrams: A Meta-Analysis Study	2009	GS	MA	Usabilidad, Mantenibilidad	V
MDA-based tool chain for web services development	2009	GS	CE	Productividad	V
A Comparative Case Study of Model Driven Development vs Traditional Development: The Tortoise or the Hare	2009	GS	CE	Productividad	V
Linking model-driven development and software architecture: A case study	2009	IEEE	CE	Productividad	V
Model-driven development for early aspects, Volume 52, Issue 3	2010	GS	CE	Mantenibilidad, Eficiencia	Ev
Distribution of Effort Among Software Development Artefacts: An Initial Case Study	2010	GS	CE	Productividad	V
The impact of structural complexity on the understandability of UML statechart diagrams	2010	GS	E	Usabilidad, Productividad	V
MOOGLE: A model search engine	2010	GS	E	Productividad	Ev
An Empirical Investigation of the Utility of 'pre-CIM' models	2010	GS	E	Productividad	Ev
Towards to the validation of a usability evaluation method for model-driven web development	2010	GS	E	Eficiencia, Usabilidad	Ev
Productivity Analysis of the Distributed QoS Modeling Language	2010	GS	CE	Productividad	V
Usability evaluation of multi-device/platform user interfaces generated by model-driven engineering	2010	GS	E	Usabilidad	V
Improving Efficiency of Change Impact Assessment Using Graphical Requirement Specifications: An Experiment	2010	GS	E	Eficiencia, Mantenibilidad	V
Moppet: A Model-Driven Performance Engineering Framework for Wireless Sensor Networks	2010	GS	CE	Mantenibilidad	Ev
2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications (The Impact of Model Driven Development on the Software Architecture Process)	2010	GS	CE	Eficiencia	V
Enterprise systems development: Impact of various software development methodologies	2010	GS	CE	Productividad	V
From aspect-oriented models to aspect-oriented code?: the maintenance perspective	2010	ACM	E	Mantenibilidad	Ev
Generating blogs out of product catalogues: An MDE approach	2010	ACM	E	Productividad	Ev
Leyenda:					
GS: Google Scholar	CE: Caso de Estudio	S: Encuesta (Survey)			
	E: Experimento	Ev: Evaluación			
	MA: Meta análisis	V: Validación			

Tabla A.1. Mapa sistemático