Systematic review about the integration of model-driven software development and agile methodologies

Revisión sistemática de la integración de modelos de desarrollo de software dirigido por modelos y metodologías ágiles

Received: 30- 04 - 2016 Accepted: 26-05-2016

Juan José Morales Arias[1]
César Jesús Pardo Calvache[2]

Abstract

Currently, in some instances of the software development industry are carried out by means of manual activities and/or robust methodologies which can be often heavy and inefficient. This situation brings several issues related to the difficulty to produce software in a timely manner, agile, at low cost and with a high quality level. A way to improve this situation is to incorporate in the software development process the formalism and abstraction needed to automate and optimize the most critical tasks defined from methodologies used in software companies and starting from an agile approach. This would add value to the business and would improve significantly the process of software. In this sense, in order to publicize the benefits of agile approaches and programming environments driven models, a systematic review of the literature has been conducted so as to the projects where these approaches have been integrated globally. Besides, it has been possible to identify some benefits, which have been reported by different studies.

Keywords*:* Model driven software development (MDSD); model driven architecture (MDA); agile; software process; development.

Introduction

The first programming languages such as Assemble (Batsonetal., 2015), allowed to write programs in which one could move data to an specific memory address and execute operations characteristic of the processor, however,

[1] Colombian. Systems engineer. Especialist in software development from the Universidad EAFIT, with experience in software development and agile methodologies supported over SCRUM, experience in technologies such as J2EE, Oracle PL\SQL, LAMP, WEB with AngularJS, GWT and Struts environments. E-Mail:jmoral24@eafit. edu.co.

[2] Colombian. PhD and MSc. in Informatics Engineer from the Universidad de Castilla-La mancha. Assistant teacher in the Systems Department, Engineering Faculty, Electronics and Telecommunications Universidad del Cauca. Popayán, Colombia. E-mail: cpardo@unicauca.edu.co.

on this point the abstraction level was closer and distance the language in which the human problems re generally developed. After the low levels of language, more structured procedural languages arose, such as: FORTRAN (Chivers & Sleightholme, 2016) and the C language (Kernighan & Ritchie, 1988) where it was possible to define instruction sets with close names to reality, which tried to represent them and call them in a logical order inside a program structure. At the end of the 1980s, languages aimed at the objects started to distance from the tight concept relations with the computer (hardware), program thinking more in the object characteristics, actions and relations that were being analyzed (Terren, Moreno, & Jimenez, 2007). There, the abstraction language and level with which we represent the reality in a computer program had considerably evolved in comparison to its beginning.

In the early XXI century, the Unified Modeling Language (UML) was developed, modeling language that increased the abstraction level used to represent reality on informatics systems that were developed (Object Management Group, 2015). This language was integrated to the software development processes that had achieved acceptation levels in the software industry, for example, Rational Unified Process (RUP) or its agile variant Agile Unified Process (AUP), use this modeling language in their stages to represent each one of the stages. In this epoch, even tool development was held in order to generate software coding from built models (Objects by Design, 2016). Although, the generated code was too basic, and thus, the models were considered useful only in early software development stages, that is to say, in analysis and design stages, and the generated code was disposed or simply it wasn't used, since its adaptation and modification in some cases, was more expensive than developing an app from scratch (Andrade, Ferreira, & Sinderen, 2004).

In this sense, some researchers started to take interest in the implementation of methodologies focused in the using of models and their transformation, according to certain defined criteria through a domain-specific language (DSL) in executable software (Haase et al., 2007; The eclipse Foundation, 2016 ; Völter et al., 2006). In order to improve the software development industry, and increase the abstraction levels where the representative software and models are coupled and can be changes at same speed like businesses do.

But, there is not an extended Model Driven Software Development (MDSD) yet and the clear guidelines about their implementation on general level have still not been identified, since the done efforts in this sense are isolated and denote certain maturity level that has disabled its industrialization (Bernardo-Quintero & Duitama-Muñoz, 2011).

On other hand, the industry is trending to integrate and institutionalize agile and less complex software that allow them to adapt themselves to the business change in an appropriate manner in the less time possible (Fallas, 2012; Zhang & Patel, 2011). Taking in account that the previous is kindred to the MDSD objectives, in the sense that it emphasizes in the system changing regarding reality which pretends that they pretend to represent and the early delivery of valuable products, some proposals from different points of view have been made, whose goal is to take advantages of these two methodologies and formulate a development process where the best of these two focuses is applied.

The systematic review offers a perspective of the developed initiatives regarding software development, directed towards models inside the agile methodologies context, analyzing which practical applications have been held and which their results have been.

The document was organized in way that: the first part shows a detailed description of the used process to make the systematic review including the answer formulation, source and studies selection and information extraction. Then; it shows the obtained results in the study based in the analyses results. Finally, the conclusions and future work are presented.

Systematic review design

For the systematic review for the software development aimed to agile models and methodologies, the protocol template from (Biolchini, et al., 2005) and the review protocol presented (Brereton et al., 2008) were used, which was mainly focused in: (i) the question formulation, (ii) source selection, (iii) study selection, (iv) information extraction, and (v) results summarize. The Figure 1 shows a flow diagram of the held activities in the reviewing process.

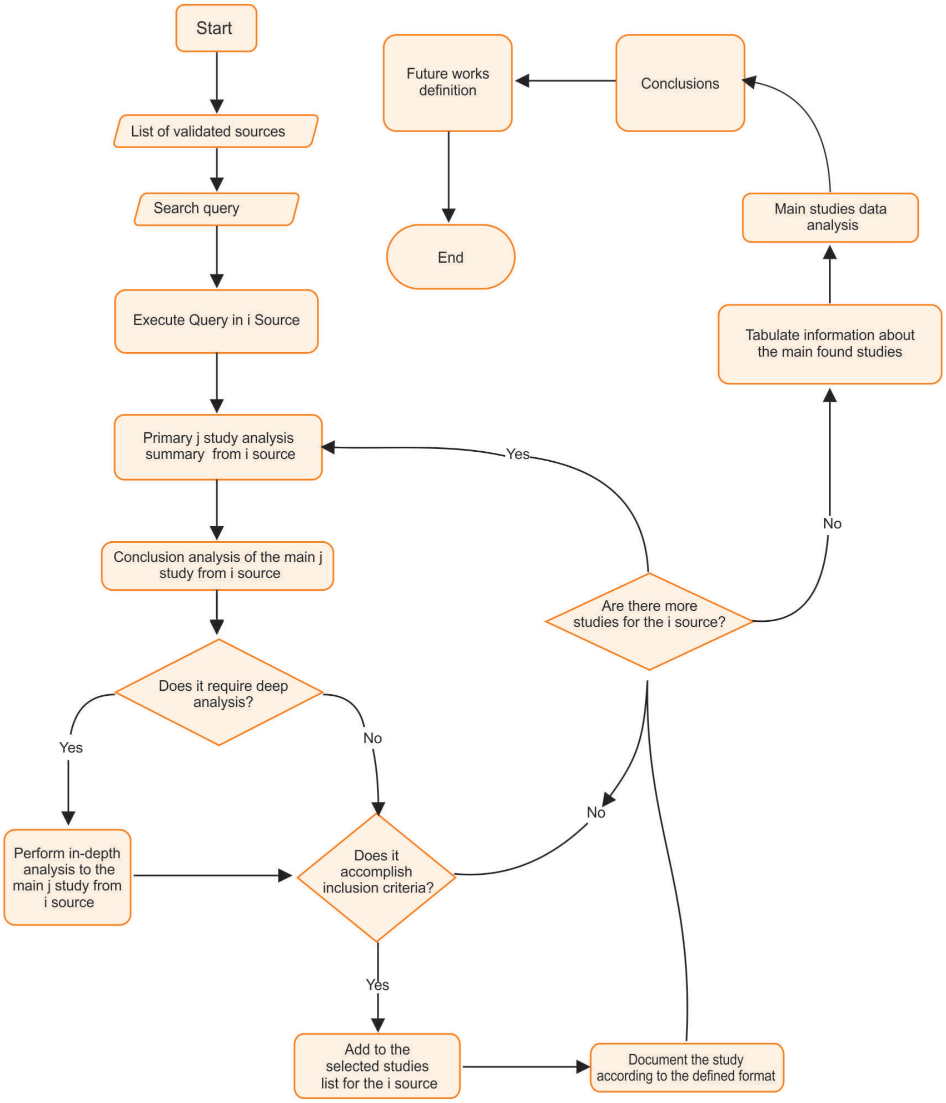
Start
List of validated sources
Search query
Execute Query in i Source
Primary j study analysis summary from i source
Conclusion analysis of the main j study from i source
Does it require deep analysis?
Yes
No
Perform in-depth analysis to the main j study from i source
Does it accomplish inclusion criteria?
Yes
No
Add to the selected studies list for the i source
Document the study according to the defined format
Are there more studies for the i source?
Yes
No
Tabulate information about the main found studies
Main studies data analysis
Conclusions
Future works definition
End

Figure 1. Systematic review protocols
Source: The authors

Research question formulation

The research question defined by the performing of this systematic review was: Which related initiatives and Jobs are related with the software development aimed to agile models and methodologies have been executed? The Chart 1 shows a list of terms used to design the research question.

In the context of the planned systematic review, the related proposals and existent works were observed and analyzed with the software development directed by models and their application in agile environments. As a control mechanism, the related works were reviewed, among them are books that deepen the practice application and some activities held in the generic formation. Of these works key words have been obtained, and despite that most of these works have not been found in the selected information search sources, it has been considered important to have them as a reference in a set of main studies as gray literature due to its relevance and relation with the systemic review.

The population under study in the systematic review was presented in the selected sources data and they were related to the systematic review. In chart 1 the terms used are shown.

Chart 1. Used terms

Terms	
Key words	Synonym
Process	Procedure, developed, technique
Driven	Guided,based,oriented
Model	pattern, type, archetype
Software	System
Agile	Rapid
Activities	Action, steps
Architecture	Design, definition, structure

Source: The authors.

Source selection

With the listing of key words and the use of "AND","OR","NOT" logical connectors, the following search strategies were held:

(process OR procedure OR techniques OR development) AND (model OR patterns OR template) AND (architecture OR design OR structure) AND (driven OR guided OR based OR oriented).

AND (software OR system) AND development AND (activities OR actions OR steps) AND (agile OR scrum OR xp).

In the systemic review the following sources were used: Wiley Online Library in the topic ofComputer Science, ACM Digital Library, ProQuest and the IEEE Computer Society, the official sites of Manifiesto Ágil, SCRUM, eXtreme Programming (XP), and other related articles with the topic were reviewed as grey literature. In the moment of performing the strategic search, the characteristics of each search engine were adapted from the chosen sources.

Studies selection

Defined as the main information sources, the selection and analysis process was described by the search results, in order to determine their relevancy and contribution.

The study selection was based in an incremental iterative process, which allowed to perform the results search and extractions of each one of the sources. The incremental process was developed in succession or iterated in each one of the search sources, thus, the information increased as the resulting values were added to the chosen studies.

Information extraction

The main studies inclusion criteria, are mainly focused in the analysis of the following perks: (i) title, (ii) summary, and (iii) the conclusion of each study. It is looking to identify in which measure these were proposed or reviewed for the implementation of process developments aimed towards agile development methodologies. To determine the relevance of the articles, it was necessary to make an in-depth analysis of their content, taking in account variables such as proposed model description, its relation with the agile methodologies domain, and its pragmatism.

An exclusion criteria, it was taking in account, after an in-depth review, the articles that made a general of the methodology study or diagnose, but didn't define or proposed an application model in a concrete way.

The documentation of the main selected studies, it was structured in a chart in order to define the concrete interest variables in which it was taken in account the following information: code, title, subtitle, source, country, year, authors, key words, references (secondary studies), main ideas, proposed process type and applications. The chart 2 shows a summary of the used format to represent the relevant information.

Chart 2. Example of the template used to organize the information of a primary study

Title	Towards Patterns for MDE-Related Processes to Detect and Handle Changeability Risks
Subtitle	No info.
Source	ACM
Country	GERMANY
Year	2012

Continued on next page

Continued from previous page

Authors	Regina Hebig, Gregor Gabrysiak, and Holger Giese
Key words	Software development process, analysis, control change, patters

Main Ideas

In this article, a detailed study is held about the patterns that can be applied in cases where the developed models under the MDSD methodology have repeated changes overtime. Different forms are posed in which the system integrity can be affected and a notation is suggested to represent the different cases in which this risks can be presented, besides of 4 possible applications that allow to mitigate them.

The software development process has a direct impact in the software productivity and product quality. Likewise, the best productivity and quality obtained through the Model Driven-Engineering (MDE) have to be the reflection of a good model that supports the software development dynamic in the context of business context that easily change overtime.

Three MDE stages were discovered in which it can control the system's changing capacity:

- Artifact content loss or preservation.
- Explicit relation losing between artifacts.
- Automatic activities that can end up hooked so that two changes can't be given independent of the other.

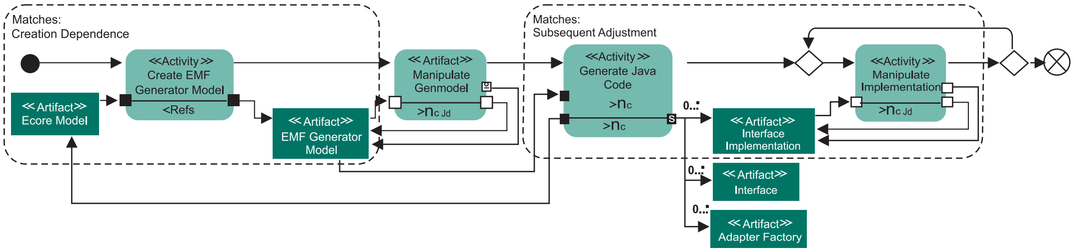

Figure 3. Exemplary process of the EMF case study with initially perceived SwMamo activities

The next pattern set is also posed in order to handle each situation:

1. 1. Process Proto Pattern Subsequent Adjustment: Is used in the unexpected content loss. The initial automatic form of creation activity creates and adjusted artifact and fills it with content in the base of the main artifact (that is to say, as well as the artifacts tab in content after the execution). This way, there is no version that is considered an adjusted artifact. The activity adjustment adds even more to the manually adjusted artifact content.

2. Process Proto Pattern Creation Dependence: Este es un antipatrón que hace referencia a la dependencia de referencias entre artefactos. Donde un artefacto de entrada es a la vez un artefacto de salida, creando así una dependencia cíclica.

3. Process Proto Pattern Split Manufacture: Este patrón hace referencia al caso de pérdida inesperada de contenido en un artefacto. Aquí se propone conservar el detalle de artefacto antes y después de ser regenerado, y ajustar manualmente los detalles de este.

4. Process Proto Pattern Anchor: Abarca en el caso de pérdida de relaciones explicitas entre artefactos. Aquí cada referencia tiene un ancla en el artefacto que guarda la referencia correspondiente a este en cada regeneración o cambio manual.

Proposed process type

A set of activities, patterns and anti-patterns are proposed in order to cover the given risks, due to the change capacity in an MDE system context.

Applications

A study case with the SAP development team using EMF is done, comparing the proposed activities and patterns.

Source: The authors

Results and discussion

With the extracted information an analysis of the obtain data was held, taken in account the general and specific results after these had been applied in the inclusion and exclusion criteria.

Publishing tendencies

After applying the protocol to obtain the main studies, 42 relevant studies related with the development guided by models were found, in which 22 of them a specific relation with the MDSD processes had, aimed towards the agile methodologies. After a detailed analysis, the 10 main studies were selected to be reviewed under inclusion and exclusion criteria, where it was evaluated in which measurement they proposed an specific process for the application of MDSD in agile development environments. The Figure 2 shows the publishing tendencies since the year 2000 until 2014. As it can be seen, there is an increasing interest in the MDSD area in the last decade, focusing a main interest between the years 2006 and 2012, where 50% of the related found studies in relation with this topic. This indicator allows to establish that there is an interest in the software development market and academy, and theoretical and practical works are being advancing. On its part, we haven't found relevant studies for the year 2015, maybe, because the first trimester of the year is not over yet.

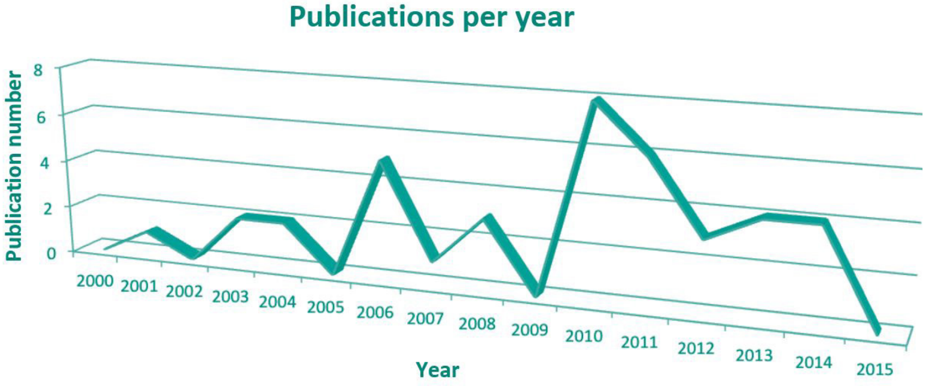

Figure 2. Publications per year tendency
Source: The authors.

Country distribution

The Figure 3 shows the distribution in descending order of the countries that show the most performed studies in relation to the MDSD, these being: USA, Brazil, Spain, Germany and China. In relation to Brazil, China and Kenya, they show interest in the formalization of the software industry, that besides, have performed meaningful studies that cover the MDSD specific topic, for example, in Kenya an Adaptable MDA was developed for formal methods and its integration with agile methodologies (Rigworo, 2013), in Brazil aprocess was developed directed by the model aimed towards testing (Almeida & Oliveira, 2014), and in China a conceptual focus was designed to model a software model development directed by models (Duan & Fu, 2006).

The mentioned allowed to observe the various points of view related to the developed proposals and solutions, and this brings along a wide range of possibilities that can be implemented in different types of businesses related with software development, where the generic software architecture was applied for different business domains, and where there is a scope to the continuous product delivering. Likewise, it allows to observe that even emerging industries are looking at the MDSD as an applicable option for their contexts. On other side, the in the origin of main simple studies that this scope is acquiring more and more interest in the software industry, and that it is not a remote tendency in certain privileged development and research sectors, for example:

There are tools for the generation of we apps through models that link with business processes and that have been successfully implemented in entrepreneurial environments (Kraus, Knapp, & Koch, 2007; Kroiss, Koch, & Knapp, 2009).

ETL generators have been implemented (Extraction processes, data analysis transformation and load) inside the business intelligence processes, which allow, from a model data base, to generate data extractions to create and populate an intermediate model with a base capable of consulting that contribute valuable information for the choice-making (Oracle, 2015; El Akkaoui, Zimànyi, Mazón, & Trujillo, 2011).

There are tools that generate graphic Web interfaces drawing from model, that at the same time allow to manage data models generated through business models. In this case we have, for example, Open Xava (Paniza, 2011), Moskitt (Hernández, León, & Ferrara, 2015), Spring Roo (Sarin, 2011), among others.

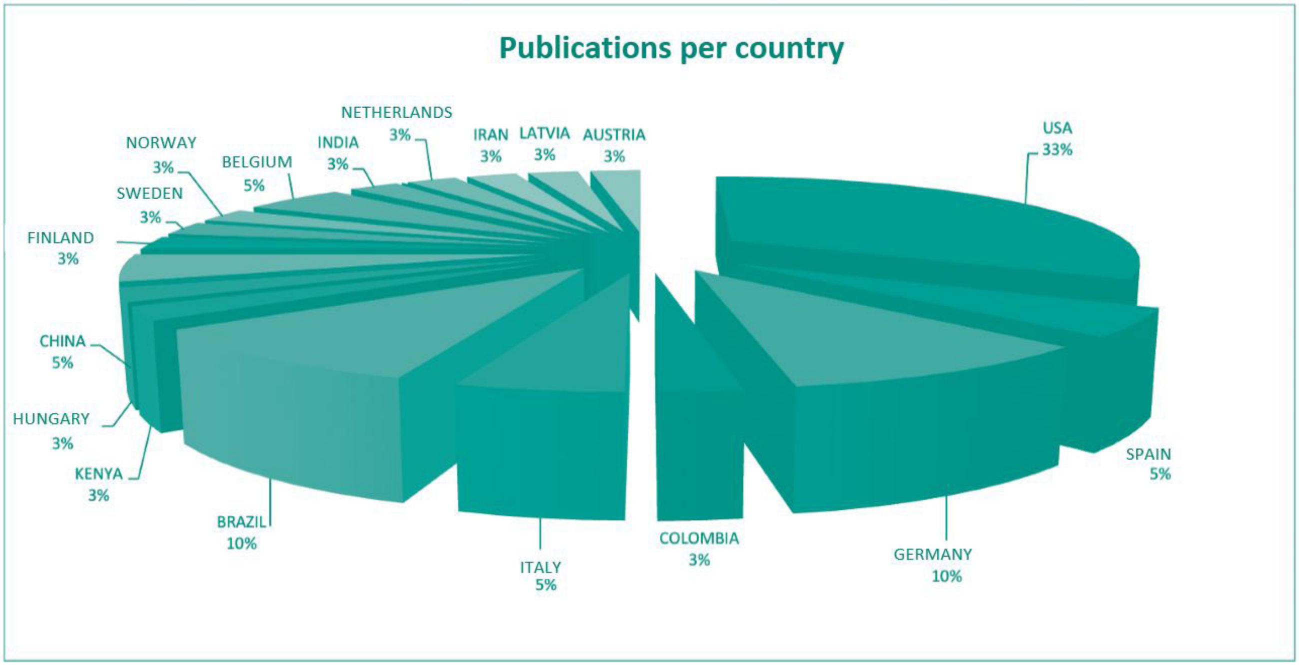

Figure 3. Publications per country.
Source: The authors.

It is important to highlight that the strengthening of the software industry depends directly in the help of the State, universities and businesses. In Colombia, the Ministry of Information and Communication Technology (ICT) (MinTIC, 2016), and other national and departmental organisms such as the SENA (SENA, 2016), INNPULSA (INNPULSA, 2016), RED CLUSTER Colombia (RED CLUSTER COLOMBIA, 2016), (Ruta

N, 2016), Parquesoft (Parquesoft, 2016), Bancoldex

(Bancoldex, 2016), Fedesoft (Fedesoft, 2016) (Market, 2012), Intersoftware (Intersoftware, 2016), clusters, in which the following stand out: PacifiTIC (PacifiTIC, 2016), CaribeTIC (CaribeTIC, 2016), SinerTIC (SinerTIC, 2016), ClusterTIC Antioquia (ClusterTIC, 2016), Apps.co (Apps.co, 2016), Vive Digital, commerce chambers, among others, have encouraged initiatives that have allowed to strengthen the ICT sector in Colombia. However, the addressed subjects driven by models in Colombia are still very distant, maybe, due to the fact that the benefits of such is not known.

Identified benefits of the MDSD implementation in businesses

Drawing from the main studies analyses, it has been possible to identify some benefits in the implementation of software development driven by models. The Chart 3 displays a set of identified benefits.

Chart 3. Benefits of the application of MDSD inside the software development processes

	Benefit	Description
1	Quality	The software generated under the DSL will have the same reference architecture and will not be exposed to human mistakes inherent to the complexity of the code that would have to be hand written, in case the process is done manually
2	Code migration and generation in multiple languages-Portability	A product can be easily migrated to the newer technology version from which it was implemented or to a different technology. This is possible, thanks that in this case it is only needed to change the Model to Text (M2T) transformation mechanisms without affecting the model. Thus, a product that is currently being implemented in a PHP language can be easily migrated to other languages, for example, Ruby or Python only changing the transformation for each specific language.
3	Integration	Drawing from the implementation model, one can automatically generate a model that at the same time generates an automatic executable unitary test set. Equally, continuous integration processes can be modeled, in which the procedures are defined in order to deploy the app in a specific environment.
4	Productivity	Although initial efforts will focus in having a specific language thick enough to handle the business needs, this focus allows to gradually increase the team development productivity, improving the delivery time of each of the defined deliverables. This because, what before needed more time to be coded, now will be auto generated through the M2T generations.
5		Through automation one can generate an executable code from formal models, using one or more transformation stages.
6	Scope separation	The system aspects that cannot be easily treated with one module, can vary and affect the modules that depend on it. Through this scope the redundancy of the generated code decrease including its dependencies.
7	Reusing	Once having the DSL defined corresponding to the architecture and business, they can be reused in different applications that use the same structure or are related with the business field.
8	Complexity control through abstraction	The DSLs allow to take problems to a more basic complexity level and closer to the real world, encapsulating their technical complexity the transformation processes that only have to be done once.
9		The MDSD follows the same guidelines as the OMG (Object Management Group) [46], where the model independence and standardization is established. This allows to inter-operate with other technologies that follow the same standards.

Source: The authors

Proposed solutions

The Chart 4 shows the classification of the proposed processes according to their focus and relation with conventional software development methodologies. Likewise, it could be observed that in 50% of the main studies processes are proposed or applied in which the integration processes or the engineering practices guided by the models or the MDE and agile methodologies are carried out (Cockburn, 2006 ; Canós, Letelier, & Penadés, 2003). It is also observed that the MDSD has been joined with the RUP (Kroll, Kruchten, & Booch, 2003), where the goal was to integrate the MDE activities to the RUP design and elaboration stages (Kroll et al., 2003). Other Solutions are aimed in integrating MDT methodologies inside an incremental and spiral process (Almeida & Oliveira, 2014), or to look to control efficiently and through patters the model changes that can affect the generated model in different stages (Hebig, Gabrysiak, & Giese, 2012) and in model to model (M2M) methodologies (The Eclipse Foundation, 2012), or in the layer and levels generation of code abstraction drawing from the same model (El Akkaoui et al., 2011).

On Chat 5 is displayed the general analysis level in the found proposals in the analyzed studies, its relation with other focuses and methodologies, and a small description of their advantages. Starting from the analysis tendencies and focuses were identified in the proposed processes. In first place, there is a strong tendency in comparing the MDE practices, including its activities and principles to formal processes as RUP, Scrum and XP. In this tendency, processes such as MDD-SLAP stand out(Zhang & Patel, 2011), just as in the agile management methodology, Scrum, is also formed by sprints, with an iteration set that describe activities original of the MDE or as well other software engineering areas, as a quality assurance and continuous integration. On other side, there are the SAGE processes (Matinnejad, 2011), whose main objective is to apply agile focuses, bearing the delivery of executable codes coming from conflictive modules. This process uses a set of different focuses that define different system aspects, such as behavior, reaching and interactions, and from the transformation of these models, to generate an executable code.

As it could be seen on Chart 3, there is decrease in the traditional software development methods, which require a great amount of hand-made documentation (including the coding). Yet, as displayed on Chart 4, when organizing the studies by proposed type, it is possible to find that some proposals are based on processes like the RUP and other formal methods, these focus in mapping the corresponding phrases to the implementation with activities from the MDSD.

Chart 4. Proposed process type percentage

Qty.	Focus	Publications	%
1	Spiral MDA - MDT	1	10%
2	MDE Patterns Changing handling	1	10%
3	Multilayer transformations	1	10%
4	Unification with agile methodologies	5	50%
5	Unification with RUP	1	10%
6	DSL Application to formal methods	1	10%

Source: The authors.

Chart 5. Main proposed processes

Process name	Aiming	Main Objective	Advantages	Year
SAGE (Kirby, 2006)	Based in MDD.	Applies highly agile software orienatation orientación ágil orientation to highly ensured software.	Bears partially conflictive generación de executable models generation.	2006
Hybrid MDD (Guta, Schreiner, & Draheim, 2009)	Based in Assembly.	Applies agile orientation orientación ágil for medium and proyectos medianos small projects.	Partial application of parcial de MDD activities in collaboration with colaboración con traditional programming tradicionales practices.	2009
MDD –SLAP (Zhang & Patel, 2011)	Based in Agile.	Benefits from the agile methodologies and MDD advantages in order to develop real time communication systems	Establishes a correspondence between the MDD and the agile practices	2011
High Level Lifecycle (Ambler, 2004)(Ambler, 2002)	Based in Agile.	Escalates the agile development	Great level of appliable agile MDD.	2004

Fuente: los autores.

Conclusions and future Works

The systematic review held in this work, gathers the results of the main studies done with agile processes for the software development directed by models. The review has followed a formalism that has allowed to obtain a complete topic perspective in the study subject. And also to validate the obtained results from the defined protocol to carry out the systematic review.

The selected studies through defined inclusion and exclusion processes allowed to obtained results related with the study subject. Through said results, it was possible to identify the interest for the software development guided by models such as the adaptation of the focus and/or development processes with agile aiming. With the identifying of the increasing interest for the integration of agile methodologies along with focuses guided by models, it is possible to think about the impact of these two focuses in the micro, small and medium businesses isn't very far from the obtained benefit by big businesses that use the MDSD.

Related to the analyzed proposals it is necessary to stand out that each one of these mentioned cover a specific aspect from the development process as well as its agile approaching and application in the software industry in a big scale. However, there are some aspects that are not clear, for example:

Is it possible to integrate software development stages, tests and deployment as requirements inside an MDA frame?

Is it possible to map the same proposed processes from maturity and CMMI software development models?

Which roles interfere in a process in which an MDE can be integrated with agile methodologies? Should new roles be integrated? Which ones and with which stage?

Are there tools that support the integration of these methodologies and all of the process stages?

Is it possible to apply these kind of methodologies in small and medium businesses in a practical form and at low cost?

These set of emerging questions, allow us to conclude that, there is actually a great interest and meaningful developments for the process formulation and application of software development aimed by the models, these are not yet in a maturity stage, and it is necessary to propose Solutions with the goal to cover a wider set of aspects in the software development at industrial level, which allow to increase the quality and productivity levels in the software development making joint and synchronized use of the MDD methodologies in the context of agile development processes. As mentioned at the beginning of the results analysis, no relevant studies in the year 2015 haven't been found, maybe, because the first semester of the year 2016 is not over yet and it is of supposing that the related Works haven been indexed yet by the consulted sources.

The industrialization is one of the necessary development processes, and it is expected it is not far from becoming a possible stage as other type of indutries. It is important to recognize the MDSD benefits, as well as the effort to perform the solution definition, adaptation and implementation, and strategies that allow the productive processes in the software industry, more than only doing it at management level as it is done nowadays through the worldwide certified model recognized as CMMI. The MDSD can be used as a solution to give support to the accomplishing of good practices defined by other models, such as CMMU, agile methodologies, among other. There is work to do to reach superior maturity levels hat allow to have a refined and productive process, and at the same time cover different critical aspects in the software development, such as quality, change resistance and scalability. From the work done in this sense, it depends how much the industry separate from the handmade processes that affect their effectiveness, opportunity and resistance.

From the obtained results done in this systematic revision, as well as in future works, it is expected to cover two lines of work. In the first place, it is expected to carry out a second protocol execution in new information sources, this with the goal of identifying a higher number of works that were not found in the treated sources in this revision. On second place, to design a proposal that allows to conduct to the software development of MSB businesses through the ingestion of the two analyzed scopes (i) MDSD (ii) agile focuses. With this solution it is expected to easy the software development based on model and through an agile focus in MSB developers of the software.

Acknowledgements

César Pardo greets the contribution to the de la Universidad del Cauca where he works as an assistant teacher.

References

Almeida, C. C. de J., & Oliveira, A. A. de. (2014). Qualitas: A Proposal of Process Model Development Software Driven Models. In *Proceedings of the 7th Euro American Conference on Telematics and Information Systems* (pp. 28:1–28:4). New York, NY, USA: ACM. doi:10.1145/2590651.2590678

Ambler, S. (2002). *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process.* New York: John Wiley & Sons, Inc.

Ambler, S. (2004). *The Object Primer: Agile Model-Driven Development with UML 2.0*. Cambridge University Press.

Andrade Almeida, J. P., Ferreira Pires, L., & Sinderen, M. J. van.(2004). Costsandbenefitsofmultiple levels of models in MDA development. In *Second European Workshop on Model Driven Architecture with an Emphasis on Methodologies and Transformations* (pp. 12–20). Canterbury, UK.

Apps.co. (2016). *Mobile apps encouraging*. Recovered May 5th, 2016, from https://apps.co/

Bancoldex. (2016). *Entrepreneurial Colombian development bank*. Recovered May 5th, 2016, from https://goo.gl/FDHOJY

Batson, A., Lack, M., & Jones, A. (2015). *x86 Assembly Guide*. Recovered from http://goo.gl/LMU44

Bernardo-Quintero, J., & Duitama-Muñoz, J. F. (2011). Reflections about the adoption of centered models in software development. *Ingeniería Y Universidad*, *15*(1), 219–243.

Biolchini, J., Gomes, P., Cruz, A., & Travassos, G. (2005). *Systematic Review in Software Engineering*. Rio de Janeiro, Brazil: Systems Engineering and Computer Science Department, UFRJ. Recovered from http://goo.gl/ymUlwp

Brereton, P., Kitchenham, B., Budgen, D., & Li, Z. (2008). Using a Protocol Template for Case Study Planning. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE 2008)* (pp. 1–8). Bari, Italy.

Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Agile Methodologies in the Software Development Ágiles en el Desarrollo de Software. In *VIII Software Engineering and Database Conferences (SEDC 2003)* (pp. 1–8). Alicante, España.

CaribeTIC. (2016). *Caribbean in the global CIT, the CIT in the regional economic map*. Recovered May 5, 2016, from http://www.caribetic. com/

Chivers, I., & Sleightholme, J. (2016). *Introduction to Programming with Fortran*. London, UK: Springer London. doi:10.1007/978-0-85729-233-9

ClusterTIC. (2016). *Information, Communication and Technology Cluster*. Recovered from http:// goo.gl/ CzPrIC

Cockburn, A. (2006). *Cockburn: Agile Software Development: The cooperative Game*. Addison-Wesley Professional.

Duan, Y., & Fu, X. (2006). A Conceptual Approach to Modeling Model Driven Development Processes. In *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)* (pp. 179–198). Morne, Mauritius.

El Akkaoui, Z., Zimànyi, E., Mazón, J.-N., & Trujillo, J. (2011). A Model-driven Framework for ETL Process Development. In *Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP* (pp. 45–52). New York, NY, USA: ACM.

Fallas, J. M. M. (2012). Agile methodologies applied in the Project Management *Software*. Valencia, España. Recovered from http://goo.gl/EfQF4x

Fedesoft. (2016). *Colombian Federation of Software Industry*. Recovered from http://fedesoft.org/

Guta, G., Schreiner, W., & Draheim, D. (2009). A Lightweight MDSD Process Applied in Small Projects. In *35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2009)* (pp. 255–258). Petras, Greece. doi:10.1109/SEAA.2009.63

Haase, A., Völter, M., Efftinge, S., & Kolb, B. (2007). Introduction to open Architecture Ware 4.1.2. In *Model-driven development tool implementers forum (MDD-TIF'07)*. Zurich, Switzerland.

Hebig, R., Gabrysiak, G., & Giese, H. (2012). Towards patterns for MDE-related processes to detect and handle changeability risks. In *International Conference on Software and System*

Process (ICSSP 2012) (pp. 38–47). Zurich, Switzerland.

Hernández, M. G., León, B. B. P. de, & Ferrara, J. M. (2015). *MOSKITT, a free tool which supports application development methodologies*. Valencia. Recovered from http:// goo.gl/JF8SaI

INNPULSA. (2016). *Organism for suporting and encouraging extraordinary entrepreneurial growth*. Recovered from https://www.innpulsacolombia.com/

Intersoftware. (2016). *National Information Software and Technologies Federation*. Recovered from http://goo.gl/UFrNHQ

Kernighan, B. W., & Ritchie, D. M. (1988). *The C Programming Language (2nd edition)*. Prentice Hall.

Kirby, J. (2006). Model-Driven Agile Development of Reactive Multi-Agent Systems. In *30th Annual International Computer Software and Applications Conference (COMPSAC'06)* (Vol. 2, pp. 297–302).

Kraus, A., Knapp, A., & Koch, N. (2007). Model-Driven Generation of Web Applications in UWE. In *3rd International Workshop on Model-Driven Web Engineering (MDWE 2007)*. Como, Italy.

Kroiss, C., Koch, N., & Knapp, A. (2009). WE4JSF - A Model-Driven Generation Approach for Web Applications. In M. Gaedke, M. Grossniklaus, & O. Díaz (Eds.), *LNCS: Web Engineering: 9th International Conference, ICWE 2009 San Sebastián, Spain, June 24-26, 2009 Proceedings* (pp. 493–496). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kroll, P., Kruchten, P., & Booch, G. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP*. Addison-Wesley Professional.

Market, D. (2012). *Fedesoft encourages the software and IT productivity.* Recovered from http://bit.ly/PnxtRo

Matinnejad, R. (2011). Agile Model Driven Development: An Intelligent Compromise. In *9th International Conference on Software Engineering Research, Management and Applications (SERA 2011)* (pp. 197–202). Baltimore, USA.

MinTIC. (2016). *Communication and Information Technologies Ministry*. *CIT Sector*. Recovered from http://www.mintic.gov.co/

Object Management Group. (2015). *Unified Modeling Language - UML 2.5*. Recovered from http://www.omg.org/spec/UML/

Objects by Design. (2016). *UML products by product*. Recovered from http://www.objectsbydesign.com/tools/umltools_byProduct.html

Oracle. (2015). *Oracle Data Werehousing Guide: Overview of Extraction, Transformation, and Loading*. Recovered from http://goo.gl/brp9zB

PacifiTIC. (2016). *The CIT Cluster TIC In the Colombian Pacific*.Recovered from http://goo.gl/EYKZCm

Paniza, J. (2011). *Learn OpenXava with examples*. CreateSpace Independent Publishing Platform.

Parquesoft. (2016). *Official webpage*. Recovered from http://parquesoft.com/

RED CLUSTER COLOMBIA. (2016). *Cluster networks in Colombia*. Recovered from http://redclustercolombia.com/

Rigworo, V. K. (2013). Adaptable model-driven engineering for formal methods integration with agile techniques for design of software systems. *Academic Research International*, *4*(1), 446–456.

Ruta N. (2016). *Official webpage*. Recovered from http:// www.rutanmedellin.org/

Sarin, A. (2011). *Spring Roo 1.1 Cookbook*. Packt Publishing.

SENA. (2016). *National Learning Service. Educational offer portal.* Recovered fromhttp:// oferta.

senasofiaplus.edu.co/sofia-oferta/ inicio-sofia-plus.html

SinerTIC. (2016). *Sinertic Alliance Association*. Recovered from http://sinertic.org/

Terren, J. R., Moreno, D. A., & Jiménez, J. B. (2007). *Programming Aimed at Objects*. Marcombo S.A.

The Eclipse Foundation. (2012). *Model to Model Transformation – MMT*. Recovered from https://goo.gl/s8H5Xk

The Eclipse Foundation. (2016). *Eclipse Modeling Framework (EMF)*. Recovered from http://www.eclipse.org/modeling/emf/

Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S., & Czarnecki, K. (2006). *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons.

Zhang, Y., & Patel, S. (2011). Agile Model-Driven Development in Practice. *IEEE Software*, *28*(2), 84–91.