QuEF: An environment for the assesment of MDWE methodologies

F. J. Domínguez-Mayo, M. Mejías, M. J. Escalona, A. H. Torres
Departamento de Lenguajes y Sistemas Informáticos
University of Seville
Seville, Spain
fjdominguez@us.es, risoto@us.es, mjescalona@us.es, arturohtz@gmail.com

Abstract— **This paper presents QuEF (Quality Evaluation Framework), an environment for the assesment of Model-Driven Web Engineering (MDWE) methodologies. This approach is oriented to evaluate, through objectives measures, the quality of MDWE methodologies in a specific environment. Given the high number of methodologies available and proposed in the last years, it has become necessary to define objective evaluation tools to enable development teams to improve their methodological environment and help designers of web methodologies design new effective and efficient tools, processes and techniques. Since methodologies are constantly evolving, the need may arise not only to evaluate the quality but also to find out how it can be improved and how the quality improvement process could be optimize in order to reduce costs. Besides, an example of application to the NDT (Navigational Development Techniques) methodology is presented and the Functionality of the NDT methodology is evaluated in terms of MDE and Maturity characteristics.**

Keywords: Quality; Model-Driven Web Engineering; Methodology; Approach; Quality Assurance;

I. INTRODUCTION

Nowadays an interesting concern in development teams of web applications is how to build web application automatically, cheaper and with the best quality. In this line, the Model-Driven Engineering (MDE) paradigm to software development focuses on creating models, or abstractions, more close to some particular domain concepts rather than computing concepts. It is meant to increase productivity by maximizing compatibility between systems, simplifying the process of design, and promoting communication between individuals and teams working on the system. The rise of this paradigm already has had an influence on current web developments. The most important research initiative in this area is the Model Driven Architecture (MDA), which is Model Driven Architecture being developed under the umbrella of the Object Management Group (OMG)[1]. According to the OMG models may have the quality of being independent from the characteristics of any technological platform. Besides, the lifecycle of a software system is completely covered, starting from requirements capture, passing through the generation of code, and up to the system maintenance.

In this context, Web Engineering is a specific domain in which Model-Driven software development can be successfully applied. The use of MDE in Web Engineering is called Model-Driven Web Engineering (MDWE)[4] and, as can be noticed through different papers [14], [9]. In the last years several research groups have proposed different methodologies with processes, models and techniques to build applications as UWE, WebML, OOHDM, WSDM, Hera or NDT. Some of them cover most of the levels of abstraction and even have tools that support the automation of transformations in the processes of development. There is a need for the suitable design and the improvement of MDWE methodologies and effective tools. To this end, our work concentrates on evaluating and comparing existing proposals although the framework could be extended in the future to other area. A tool support which may help to organizations to carry on a quality continuous improvement. In order to offer a suitable environment for evaluating quality of approaches, this paper proposes QuEF (Quality Evaluation Framework) [3], an environment for the quality evaluation of MDWE methodologies.

The paper is organized as follows. In Section II the related work is presented. Section III presents the concepts such as MDWE methodology and framework are explained and a short description of the framework components is given. In Section IV, the Quality Model component in QuEF as the main component of the framework is described and the steps for the definition of this component, its structure and process to design it are shown. In Section V, an example of applying QuEF is proposed with the NDT methodology is performed and results of the influences of Maturity and MDE characteristics in terms of Functionality quality factor are obtained. Finally, in Section VI, a set of conclusions and contributions is laid out, and possible future work is given.

II. RELATED WORK

Web quality evaluation method and some aspects of its supporting tool is discussed in [11], where a quantitative evaluation strategy to assess Web sites and application quality is defined. As far as Web metrics quality is concerned, in [1] some important metrics proposed for Web information systems are classified, with the aim of offering the user a global vision of the state of the research within this area. As far as quality evaluation is concerned, in [13], is developed and analyzed a computational model based on the concepts of

1. http://www.omg.org

the Fuzzy Logic. According the authors, it will allow the calculate the value of Usability to final user's of software developments based on ISO/IECE 9126-1:2.001 and based on the Fuzzy Model of Takagi Sugeno Kang [13]. The term quality model is often used to refer to a set of quality attributes (also known as quality characteristics) and relations between them. By answering "yes" and "no" to questions related to quality criteria, one may measure to what extent a quality criterion is achieved. ISO standards are set out in [8], especially the ISO-9126 series with the hierarchical model of six quality factors and subcharacteristics related to each factor. The idea of developing an MDE framework for evaluating quality has been applied in [10] and other papers of the same author. As software measurement is concerned, a metamodel for the definition of software measurement models; a flexible method to measure any kind of software entity represented by its corresponding metamodel and a software tool (GenMETRIC) that supports the framework is described in [6].

A framework for classifying the most relevant web metrics is proposed in [2] where is developed the WQM model (Web Quality Model) which distinguishes three dimensions related to web features, lifecycle processes and quality characteristics. From the methodological perspective, software measurement is supported by a wide variety of proposals, with the Goal Question Metric (GQM) method (Basili and Victor) and the ISO 15539 and IEEE 1061-1998 standards all deserving special attention. Various methodologies and numerous comparative studies exist on the MDWE [14], [12], [4], [9]. Along these lines, [14] must be considered, which specifically considers modeling concepts for their ubiquitous nature, together with an investigation of available support for Model-Driven Development in a comprehensive way, using a well-defined as well as fine-grained catalogue of more than 30 evaluation criteria.

III. QUEF (QUALITY EVALUATION FRAMEWORK) FOR MODEL-DRIVEN WEB ENGINEERING

Methodology and *framework* are some words commonly used in software engineering literature and sometimes their meanings are not clear. Therefore, a brief definition of what these words mean in this work is given together with a short explanation for each QuEF component.

In this work, an approach, or *Methodology*, is a Model-Driven proposal for the development of web applications. It may provide a set of guidelines, techniques, processes and/or tools for the structuring of specifications, which are expressed as models. Only web modelling approaches which are based on MDA in the framework are considered. In addition, a *framework* in this work is a basic conceptual structure composed of a set of elements used to evaluate, in this case, MDWE methodologies although it could be extended to other area or domain. Therefore, an environment with a set of elements based on existing literature is proposed as shown in Figure 1, where four components for the evaluation of the quality of MDWE methodologies can be seen:

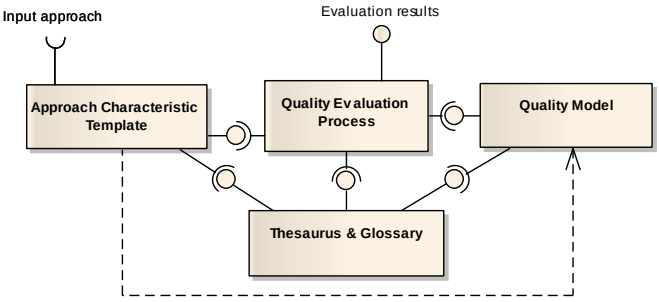


Figure 1. Component diagram of QuEF for MDWE methodologies.

Therefore, QuEF is a quality evaluation framework with a set of elements based on existing literature as shown in Figure 1, where four components for the evaluation of the quality of MDWE methodologies can be seen:

- *Thesaurus & Glossary component:* it includes all the necessary to improve the standardization of the access channel and communication between users of different MDWE methodologies.
- *Quality Model component:* it includes the basis for the specification of quality requirements with the purpose of evaluating quality. It specifies each element and its purposes.
- *Approach Characteristics Template component:* it includes the description templates of the input methodology characteristics to be evaluated. It depends of the Quality Model description.
- *Quality Evaluation Process component:* it includes the definition and specification for carrying out the quality evaluation process.

The steps for the definition of the Quality Model component, concepts, metamodel and the steps for its definition is explained in next section because it is the main component in the framework although all components are used in the evaluation example of NDT

IV. THE QUALITY MODEL COMPONENT IN QUEF

The Quality Model in QuEF is a set of characteristics, subcharacteristics and metrics, quality factors, quality attributes and the relationships between them, which provide the basis for specifying quality requirements and evaluating quality in a specific domain (in this case is MDWE). In Figure 2, the Quality Model metamodel with the relations between the different elements in the Quality model are shown, and the elements are described and explained.

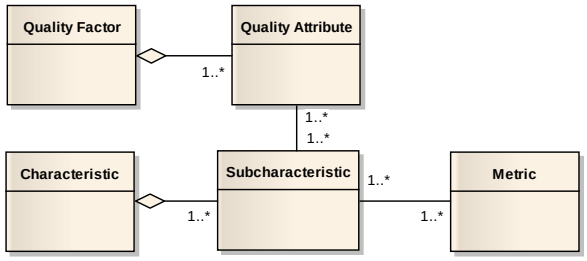


Figure 2. Quality Model metamodel

- *Quality Factor:* This is a higher-level feature that affects an item's quality. For example, a quality factor could be Usability, Functionality or Portability. Each quality factor and attribute in ISO 9126 is described in relation with a software product but in our particular case all quality factors and attributes are described in relation with approach characteristics.
- *Quality Attribute:* A quality attribute is "A feature or characteristic that affects an item's quality (Syn: quality factor)[IEEE 610]. In a hierarchy of quality attributes, higher-level attributes may be called quality factors, lower-level attributes called quality attributes" [7]. For example, Usability is defined for various quality attributes such as Learnability, Understandability, Operability, etc.
- *Characteristic*: This is a higher-level concept of an approach. It may be, for example, the software development process, models, metamodels, languages, tools, transformations or the quality assurance techniques.
- *Subcharacteristic*: This is a lower-level concept of an approach. For example, the Model-Driven Engineering characteristic may have various subcharacteristics such as, the Language Definition, Transformations and Trace Generation.
- *Metric*: In the Quality Model, metric values should indicate the degree to which a subcharacteristic is measured. In simple terms, a metric is used for measuring subcharacteristics. For example, the evaluation may be via measuring quantitatively by metrics or subjective evaluation, inspections using checklists or interviewing the users. In terms of metrics, our aim is to look for a series of qualitative and quantitative metrics based on their nature, although it might be interesting to have standard metrics on MDWE which are all, somehow, centralized. In the literature, numerous references to metrics can be found, but standardization has yet to be carried out. Furthermore, the metrics used must be validated theoretically or empirically.

Therefore, for our purposes, a *Quality Model* contains a minimal amount of *characteristics* and *subcharacteristics* through which any kind of MDWE approach can be evaluated. In order to define a *Quality Model*, it contains *association links* between the *subcharacteristics* and the *quality attributes.* These *association links* represent the dependencies between *subcharacteristics* and *quality attributes*. They show quality attributes which are affected by *subcharacteristics* or the areas of the methodology that will be significantly affected if the approach is changed. *Association links* may be based on proven and real-world experience. The impact of each *subcharacteristic* on *quality attributes* must be demonstrated and the requirements determined by real case study applications to a number of real projects. This should be supplemented by reference to published literature. Furthermore, *subcharacteristics* have to define quantitative or qualitative *metrics* which may be used to measure each *subcharacteristic*. Otherwise it would be necessary to define a set of indicators from reference values which may be set to a prescribed state based on the results of measuring or on the occurrence of a specified condition. Hence, a quality factor has various quality attributes and a characteristic has various subcharacteristics, as is shown in Figure 2. A weight is used to define the importance of a metric in the value of a subcharacteristic. Similarly, a weight is also used to define the importance of a quality attribute in the value of a quality factor and the influence importance in association links between subcharacteristics and quality attributes. The process for the definition of the Quality Model is:

- Identification of quality factors
- Identification of quality attributes for each quality factor
- Identification of characteristics
- Identification of subcharacteristics
- Identification of subcharacteristics and metrics for each characteristic
- Specification of association links between the subcharacteristics and the quality factors

A. Identification of Quality Factors

The Quality Factors of an approach include Usability, Functionality, Reliability, Maintainability and Portability. Each quality factor and attribute in ISO 9126 is described in relation with a software product whereas in our study all quality factors and attributes are described in relation with approach characteristics. In this work, Functionality is taken as an example of the quality factor. In ISO 9126, Functionality is a quality factor which is defined as: "*The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions".* This definition could be adapted to more closely fit our specific domain: *"The capability of an approach characteristic to provide functions which meet stated and implied needs when the methodology is used under specified conditions"* or in a general way could be described as: *"A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs".*

B. Identification of quality attributes for each quality factor

For each quality factor, a set of quality attributes have to be identified. For example, quality attributes related with *Functionality* are described in the same way by adapting other definitions from ISO, IEEE, other standards and work already published. Some of these quality attributes may be described as:

- *Accuracy*: The capability of an approach characteristic to provide the right or agreed results or effects with the needed degree of precision [adapted from ISO 9126]
- *Compliance*: The capability of an approach characteristic to adhere to standards, conventions or regulations in laws and similar prescriptions [adapted from ISO 9126]

- *Applicability*: The extent to which an approach characteristic is specific, useful and easy applicable for the target community.
- *Flexibility*: The extent to which an approach characteristic is expandable, adaptable and easily applied to other needs
- *Transformability*: The capability of an approach characteristic to provide an appropriate set of functions for transforming models to other models or code.
- *Testability*: The capability of an approach characteristic to enable to provide an appropriate set of functions for testing. [ISO 9126]

C. Identification of characteristics

In MDWE, models are refined progressively and transformed into new models or code with tools. Moreover, each methodology may define its development process and/or techniques. The idea is to characterize the whole MDWE process using templates. The quality of methodologies in turn depends on the following characteristics: *The Model-Driven Engineering, the knowledge of MDWE methodology users, the web modeling, the customization modeling, the maturity of a methodology, the tool support and the quality assurance techniques.*

Methodology users and developers use the available modelling languages, tools and processes and develop models based on their knowledge of the problem and their experience. A further complication is that relations are often based on judgment. For example, ISO and IEEE have different hierarchies of quality factors and attributes. Therefore, a set of general MDWE approaches, characteristics and subcharacteristics are being identified, classified and described based on experience and current literature.

D. Identification of subcharacteristics and metrics for each characteristic

For each subcharacteristic, a specification of its evaluation is necessary. For example, the evaluation may be carried out via quantitative measurement using either metrics or subjective evaluation, whereby checklists are used in inspections, or users and designers are interviewed, respectively. In this work, subcharacteristics and metrics for the Maturity characteristics are considered. For example, the Application in Real-World Projects is a subcharacteristic of Maturity. Metrics can be defined as qualitative metrics which can indicate if the subcharacteristic is *Supported (S), Partly Supported (PS)* or *Not Supported (NS)* or they can be quantitative metrics (TABLE I) but in this case, it needs an ideal expectative value to be compared. For example, in the Application in Real-World Projects subcharacteristic, a value of 50 applications could be an ideal expected value.

TABLE I. A TEMPLATE OF A SUBCHARACTERISTIC AND METRIC OF THE MATURITY

Application in Real-World Projects
Number of applications in Real-World Projects where has been applicated **= ¿?**

E. Specification of association links between the subcharacteristics and the quality factors

In this step, the association links between subcharacteristics and quality attributes are defined. A set of hypotheses are proposed to indicate which quality attribute is affected by each subcharacteristic. For example, Functionality is described as a set of quality attributes. These quality attributes could be affected by one of various subcharacteristics as shown in TABLE II.

TABLE II. FUNCTIONALITY QUALITY ATTRIBUTES AND ITS INFLUENCES IN MATURITY SUBCHARACTERISTICS AND MDE SUBCHARACTERISTICS (MATRIX OF INFLUENCES)

		Functionality quality factor					
		Accuracy	Compliance	Applicability	Flexibility	Transformability	Testability
Maturity characteristic	Topicality	X	X	X	X	X	X
	Modeling Examples	X	X	X	X	X	X
	Application in Real-World Projects	X	X	X	X	X	X
	Publications	X	X	X	X	X	X
	External Web References	X	X	X	X	X	X
MDE characteristic	Levels of Abstraction	X	X	X	X		
	Standard Definition	X	X	X	X	X	X
	Model-Based Testing	X		X			X
	Transformations	X		X		X	
	Traces	X		X			

Subcharacteristics and the relations between quality attributes are described below. Maturity and MDE subcharacteristics have different influences in the quality attributes. Our initial hypothesis is that it could bear influence on:

- *Accuracy*: These subcharacteristics considered do more accuracy the functionalities of an approach because you can get agreed results or effects with the needed degree of precision.
- *Compliance*: These subcharacteristics considered help to develop models and code easier following conventions and similar prescriptions and adhering standards.
- *Applicability*: These subcharacteristics are specifics, useful and can be easy applicable for the target community.
- *Flexibility*: Theses subcharacteristics considered can help to do more expandable, adaptable and easily applied the approaches.
- *Transformability*: Theses subcharacteristics considered in the matrix provide an appropriate set of functions for transforming models to other models or code.
- *Testability*: These subcharacteristics provide an appropriate set of functions for checking metamodels, models and code.

The Quality Model component will then be refined and improved based on results, experience or current literature. Other subcharacteristics have to be proposed and they have to

be associated with quality attributes. In this paper, a set of Maturity and MDE subcharacteristics and hypotheses for linking these subcharacteristics to quality attributes of Functionality are proposed as an example. In order to optimize costs in a quality improvement process, this influence matrix could be used to select the *minimal number of subcharacteristics which has an influence in a greater number of quality attributes*.

V. EXAMPLE OF APPLYING QUEF: NDT METHODOLOGY

A. General Description of NDT (Navigational Development Techniques)

NDT (Navigational Development Techniques) is a methodological approach oriented towards Web Engineering. It is an approach defined in the Model-Driven paradigm and it offers a suitable and easy-to-use methodological environment. With the use of NDT-Suite, NDT offers tool support for each phase of the complete life cycle of a software project. In the following evaluation of NDT, the extended revision supported by NDT-Suite is considered.

B. Applying QuEF in the NDT methodology for the Maturity Characteristic.

The Approach Characteristics Template component has been applied using an implementation in Microsoft Excel. However, the Approach Characteristics Template component has not yet been fully developed, and only Tool Support characteristic, MDE characteristic, Web Modelling and Maturity characteristic can be considered. In this example, only External Web References subcharacteristic is shown in TABLE III as an example of a template of subcharacteristic and metrics. In this type of metrics (a quantitative metric), an average value is taken as a value for External Web References on Google and other average value is taken for External Web References on Google Scholar. On the other hand, an expected value is set as an ideal value for comparing with the values in the results. Other subcharacteristics of Maturity such as Topicality, Modelling Examples, Application in Real-World Projects and Publications are not shown but they have been considered in the evaluation process of this example in next section.

TABLE III. EXTERNAL WEB REFERENCES SUBCHARACTERISTIC OF THE MATURITY CHARACTERISTIC

External Web References
NAME_APPROACH = NDT **FULL_NAME_APPROACH = Navigational Development Techniques** **Number of external web references on <u>Google</u>** Using words: *NAME_APPROACH* Approach Model-Driven Web Engineering = 1300 Using words: *NAME_APPROACH* Methodology Model-Driven Web Engineering = 1200 Using words: *FULL_NAME_APPROACH* Web Engineering = 1000 **Number of external web references on <u>Google Scholar</u>** Using words: *NAME_APPROACH* Approach Model-Driven Web Engineering = 43 Using words: *NAME_APPROACH* Methodology Model-Driven Web Engineering = 44 Using words: *FULL_NAME_APPROACH* Web Engineering = 40

For example, total metric values of Maturity and MDE subcharacteristics are shown in Figure 3. In the figure, black bars represent NDT metric values for each subcharacteristic of the Maturity and MDE characteristic respectively and grey bars represent the expectative values for an ideal approach.

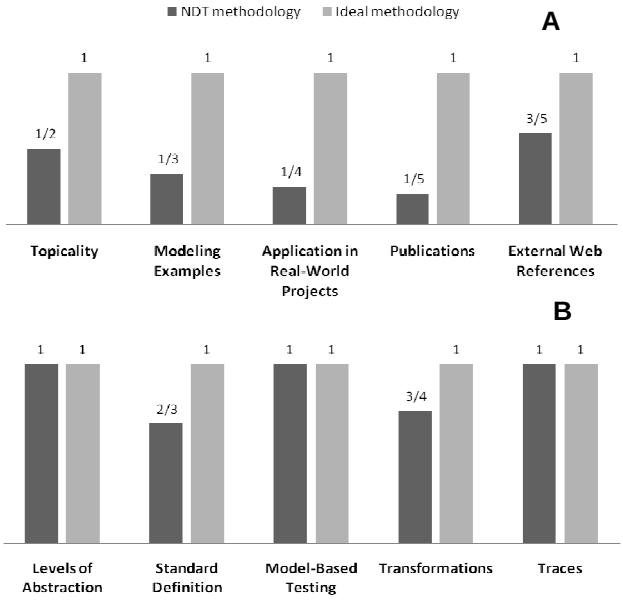
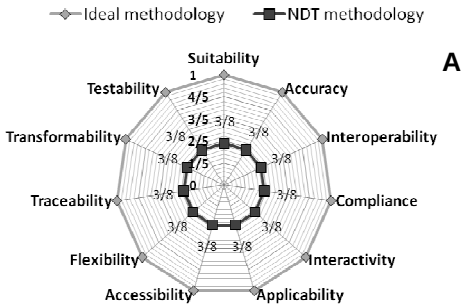


Figure 3. Different graphs which represent Maturity (A) and MDE subcharacteristic values for NDT methodology

C. An evaluation of NDT methodology

In the implementation in Microsoft Excel, Functionality, Reliability, Portability and Usability quality factors have been studied. In this example quality attributes of Functionality is shown and its relations with the MDE and Maturity characteristic. This is shown in Figure 4, where the black line represents Functionality on the NDT methodology and the grey line represents the ideal Functionality in an ideal approach according to the subcharacteristics under consideration. According to the results of the evaluation of the NDT methodology, only Maturity characteristic has been considered in A and only MDE has been considered in B for evaluating Functionality.



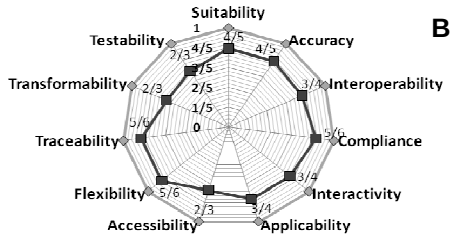


Figure 4. Functionality quality factor values for Maturity (A) and MDE (B) characteristics independently

If both characteristics are considered (for example the MDE characteristic with the Maturity characteristic) then the results could be very different. In Figure 5, the Maturity and the MDE characteristic are considered for the evaluation of the Functionality quality factor in the NDT methodology. In this case, in the graph is observed that the Maturity improves and makes more uniform the results. In these lines, we can see that NDT methodology is good in Traceability and Flexibility but it has to improve in Accessibility and Transformability.

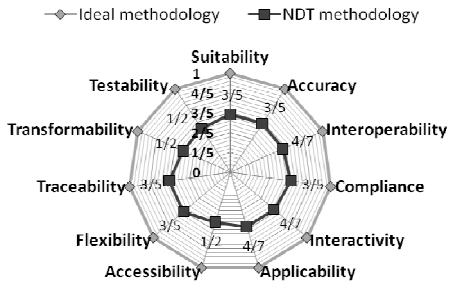


Figure 5. MDE and Maturity characteristic influences in Functionality quality factor

On the other hand, Results for Accuracy, Interoperability, Compliance, Interactivity and Applicability are uniforms for this set of quality attributes, it could be because the results are similars or because it is necessary to have more subcharacteristics and metrics for identifying differences between these quality attributes or, in other words, characteristic templates have to be defined with a major granularity of description.

VI. CONCLUSIONS & FUTURE WORK

This paper presents QuEF, a quality environment for the assessment of MDWE methodologies and it also offers an example with the evaluation of NDT methodology. Functionality quality factor have been evaluated in terms of MDE and Maturity characteristics. Furthermore, QuEF could be used for optimizing a quality continuous improvement since we can select and improve the minimal number of subcharacteristics which have a majority number of influences in quality attributes using the matrix of influences. This would help to minimize costs in the improvement of MDWE methodology environments and control it. Designers and users could have an environment where they can get a general view about the actual state of an approach in terms of several characteristics and quality factors. Besides, we have evaluated subcharacteristics related with the Maturity and MDE characteristics which are required for the measurement of the value of MDWE methodologies in order to be able to assess and improve their Functionality and defined some quality attributes of Functionality. Other MDWE methodologies as UWE and WebML are currently being evaluated using QuEF. On the other hand, it could be extended to other areas or domains. Microsoft Excel is used as a support tool although a specific software prototype is going to be developed in the future.

ACKNOWLEDGMENT

This research has been supported by the project QSimTest (TIN2007-67843-C06_03) of the Ministerio de Educación y Ciencia (TIN2007-30391-E), Spain.

REFERENCES

[1] C. Calero, J. Ruiz, M. Piattini, "Classifying web metrics using the web quality model". Online Information Review, Emerald Literari, Vol. 29, No. 3, pp. 227-248, UK, 2005.

[2] C. Calero, J. Ruiz, M. Piattini, "A Web Metrics Survey Using WQM", ICWE 2004, LNCS 3140, pp. 147–160, 2004.

[3] F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías, A. H. Torres. 2010, "Towards A Quality Evaluation Framework for Model-Driven Web Engineering Methodologies" WEBIST 2010. To be published.

[4] M.J. Escalona, G. Aragón, "NDT. A Model-Driven Approach for Web Requirements". IEEE Transactions on software engineering, Vol. 34, No. 3, pp. 377-390, 2008.

[5] M.J. Escalona, N. Koch, "Requirements Engineering for Web Applications – A comparative study". Journal of Web Engineering. Vol. 2, No. 3, pp. 193-212, 2004.

[6] F. García, M. Serrano, J. Cruz-Lemus, F. Ruiz, M. Piattini, "Managing software process measurement: A metamodel-based approach", Information Sciences.Vol. 177, No. 12, pp. 2570-2586, 2007..

[7] IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology.

[8] ISO- International Organization for Standardization, ISO/IEC 9126-1, http://www.iso.org.

[9] C. Kroiβ, N. Koch, "UWE Metamodel and Profile, User Guide and Reference". Technical Report 0802. Programming and Software Engineering Unit (PST), Institute for Informatics. Ludwig-Maximilians-Universität München, Germany, 2008.

[10] P. Mohagheghi, J. Aagedal, "Evaluating Quality in Model-Driven Engineering". International Workshop on Modeling in Software Engineering. (MISE'07), IEEE Computer Society, 2007.

[11] L. Olsina, G. Rossi, "Measuring Web Aplication Quality with WebQEM". IEEE Computer Society Press, Vol. 9 , Issue 4, pp. 20 - 29 , 2002.

[12] J. M. Pérez, F. Ruiz, M. Piattini, "Model Driven Engineering Aplicado a Business Process Management", Informe Técnico. UCLM-TSI-002, 2007.

[13] G. A. Ruiz, A. Peña, C. A. Castro,A. Alaguna, L. M. Areiza, R. D. Rincón, "Modelo de Evaluación de Calidad de Software Basado en Lógica Difusa, Aplicada a Métricas de Usabilidad de Acuerdo con la Norma ISO/IEC 9126", 2006.

[14] W. Schwinger,W. Retschitzegger, A. Schauerhuber, G. Kappel, M. Wimmer, B. Pröll, C. Cachero Castro, S. Casteleyn, O. De Troyer, P. Fraternali, I. Garrigos, F. Garzotto, A. Ginige, G-J. Houben, N. Koch, N. Moreno, O. Pastor, P. Paolini, V. Pelechano Ferragud, G. Rossi, D. Schwabe, M. Tisi, A. Vallecillo, van der Sluijs and G. Zhang, "A survey on web modeling approaches for ubiquitous web applications". International Journal of web Information Systems Vol. 4 No. 3, pp. 234-305, 2008.