Studying Maintainability on Model-Driven Web Methodologies

Francisco José Dominguez-Mayo, María José Escalona, Manuel Mejías and Jesús Torres

Abstract QuEF (Quality Evaluation Framework) is an environment to evaluate, through objective measures, the quality of Model-Driven Web Engineering (MDWE) methodologies. In this paper, this environment is presented and is used for the evaluation of the Maintainability in terms of various characteristics on MDWE. Given the high number of methodologies available and proposed over recent years, it has become necessary to define objective evaluation tools to enable organizations to improve their methodological environment and to help designers of web methodologies design new effective and efficient tools, processes and techniques and find out how it can be improved and how the quality improvement process could be optimized in order to reduce costs. This evaluation is applied to the NDT (Navigational Development Techniques) methodology, an approach that covers the complete life cycle and it is mainly oriented to the enterprise environment.

1 Introduction

Model-Driven Engineering (MDE) is a paradigm of software development which consists of the creation of models closer to a particular domain rather than concepts of a specific syntax. The domain environment specific to MDE for web engineering is called Model-Driven Web Engineering (MDWE) [9]. The Object Management Group (OMG) has developed the standard Model-Driven Architecture (MDA) which defines an architecture platform for proposals based on the Model-Driven paradigm. According to the OMG [15], the goals of MDA are portability, interoperability and

F. J. Dominguez-Mayo · M. J. Escalona · M. Mejías · J. Torres (✉)
Department of Computer Languages and Systems, University of Seville, Seville, Spain
e-mail: fjdominguez@us.es

M. J. Escalona
e-mail: mjescalona@us.es

M. Mejías
e-mail: risoto@us.es

J. Torres
e-mail: jtorres@us.es

reusability through architectural separation. The concept of platform independence appears frequently in MDA. Models may have the quality of being independent from the characteristics of any technological platform. By applying this paradigm, 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 recent years, the growing interest in the internet has led to the generation of a high number of MDWE approaches which offer a frame of reference for the Web environment [8, 12]. On the other hand, there are a high number of approaches as OOHDM, UWE or WebML without standard consensus [16] a lack in the use of standards, and scarcity of both practical experience and tool support. An example of this methodology type is the NDT (Navigational Development Techniques) which is a methodological approach oriented towards Web Engineering. It is an approach defined in the Model-Driven paradigm and it offers a methodological environment. With the use of NDT-Suite, NDT offers tool support for each phase of the complete life cycle of a software project.

There are many methodological approaches in the area of MDWE and numerous comparative studies [9, 12]. Along these lines [16], must be considered, which specifically considers modelling 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. In [3], an approach is proposed for the evaluation of Web quality that provides all the elements which are in accordance with the ISO/IEC 14598. The idea of developing an MDE framework for evaluating quality has been applied in [14] and other papers of the same author. On the other hand, in the literature there are numerous references to metrics [2, 13], according to which, software measurement integration could be achieved by adopting the MDA approach. To this end, an approach is described in [10] for the management of measurement of software processes. 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. As for web metrics quality, in [4] some useful metrics proposed for web information systems are classified, with the aim of offering the user an overall view of the state of the research within this area. With regards to the metrics model, a significant study has been revealed in [1], which proposes a set of metrics for navigational models for the analysis of the web application quality in terms of size and structural complexity.

The term *quality model* is often used to refer to a set of quality attributes (also known as quality characteristics) and the 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 [11], where particular attention is paid to the ISO-9126 series with the hierarchical model of six quality factors and subcharacteristics related to each factor.

In the light of this situation, the need to assess the quality of existing methodologies arises. In this paper, QuEF (Quality Evaluation Framework), an environment for the quality evaluation of Model-Driven Web methodologies, is proposed.

The main goal of this research is to lay the basis of an environment for the assessment of MDWE methodologies that facilitates the quality evaluation of different methodological proposals under some objective criteria in order to improve these methodologies. Hence, there is a need for the suitable design 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 cover other areas. Furthermore, the software development process has a direct influence on the quality and cost of software development, and therefore the use of an MDWE methodology and its influence on the final product quality must be considered.

This paper is organized into the following sections. In Sect. 2 QuEF is defined, the stages for the definition of the Quality Model component for QuEF are given and descriptions of every component, structure, and the process required to achieve each component are outlined. Section 3 shows an application of QuEF to the NDT methodology. Finally, in Sect. 4, a set of conclusions and contributions is laid out, and possible future work is proposed.

2 QuEF (Quality Evaluation Framework) for MDWE

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 those 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 cover other areas or domains. Therefore, an environment, QuEF, with a set of elements based on existing literature has already explained in other papers [5, 6], where four components for the evaluation of the quality of MDWE methodologies can be seen:

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

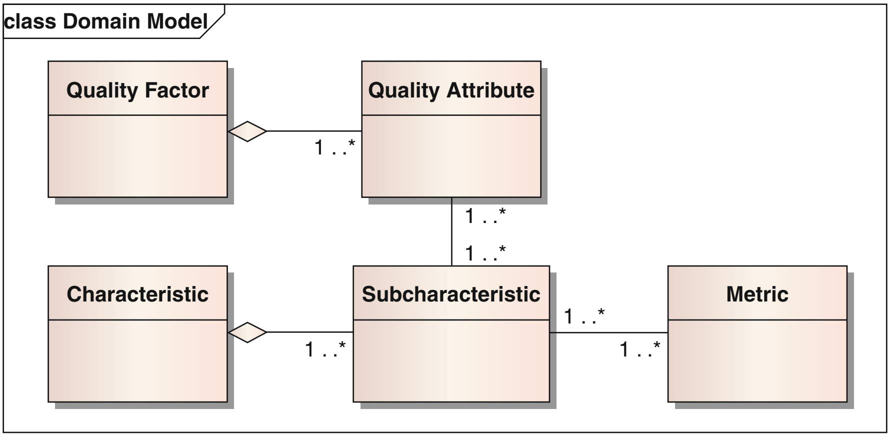

Fig. 1 Quality model metamodel

2.1 *The Quality Model Component*

The Quality Model in QuEF is a set of characteristics, subcharacteristics and metrics, quality factors, quality attributes and the relationships between these attributes, which provides the basis for the specification of quality requirements and the evaluation of quality in a specific domain (in this case, MDWE). In Fig. 1, the Quality Model metamodel with the relations between the several elements in the Quality model are shown, and the elements are described and explained.

- *Quality Factor:* This is a higher-level feature that affects the quality of an item. 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, however 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). In a hierarchy of quality attributes, higher-level attributes may be called quality factors, lower-level attributes called quality attributes". For example, Usability is defined for various quality attributes, such as Learnability, Understandability, and Operability.
- *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, metrics 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 involve measuring quantitatively by means of metrics, or may use subjective evaluation in the form of 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 establish 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.

In order to define a *Quality Model*, it must contain *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/or real-world experience. The impact of each *subcharacteristic* on *quality attributes* must be demonstrated and the requirements must be 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 Fig. 1. 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 importance of the influence in association links between subcharacteristics and quality attributes. The tasks for the definition of the Quality Model, which have already been described in other papers, are:

2.1.1 Identifying Quality Factors

A set of quality factors based on current literature, such as ISO/IEC 9126, IEEE, and other standards which are adapted to MDWE methodologies, are identified, classified and placed in a hierarchy. 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 would be described in relation with approach characteristics. In this work, *Maintainability* is taken as an example of the quality factor. In ISO 9126, *Maintainability* is a quality factor which is defined as: "*A set of attributes that bear on the effort needed to make specified modifications*". This definition could be adapted to more closely fit our specific domain: "*The ease with which a characteristic approach can be modified in order to: correct defects, meet new requirements, make future maintenance easier, or cope with a changed* *environment; these activities are known as methodology maintenance*" or in a general way could be described as: "*A set of attributes that bear on the effort needed to make specified modifications. The ease with which an approach characteristic can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment.*"

2.1.2 Identifying Quality Attributes for Each Quality Factor

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

- *Stability:* The capability of a characteristic approach to avoid unexpected effects from modifications in the approach [ISO 9126].
- *Analyzability*: The capability of a characteristic approach to be diagnosed as having deficiencies or causes of failures in the approach, or the capability of identifying those parts yet to be modified [ISO 9126].
- *Changeability*: The capability of a characteristic approach to enable specified modifications to be implemented [ISO 9126].
- *Testability*: The capability of a characteristic approach to enable a modified approach to be tested [ISO 9126].

2.1.3 Identifying Characteristics

In MDWE, models are progressively refined and transformed into new models or code. To this end, tools may also be used to test, verify or validate the models. Moreover, each methodology may define its development process and/or techniques. The quality of methodologies in turn depends on the diverse characteristics, such as the *Model-Driven Engineering*, the *knowledge of MDWE methodology users*, the *web modelling*, the *customization modelling*, the *maturity* of a methodology, the *tool support,* and the *quality assurance techniques* applied to discover faults or weaknesses. The principal idea is to characterize the whole MDWE process.

2.1.4 Identifying Subcharacteristics and Metrics for Each Characteristic

Evaluating the degree to which the quality attributes would be affected is not an easy task, and for this reason most of the metrics defined so far are qualitative metrics which indicate if the subcharacteristic is Supported (S), Partly Supported (PS) or Not Supported (NS).

Table 1 Matrix of influences between subcharacteristics and quality attributes

		Maturity characteristic					MDE characteristic					Web Modelling characteristic						
		Topicality	Modelling Examples	Application in Real-World Projects	Publications	External Web References	Levels of Abstraction	Standard Definition	Model-Based Testing	Transformations	Traces	Web Conceptual Levels	Interfaces	Development Process	Content Feature Modelling	Presentation Feature Modelling	Navigation Feature Modelling	Business Feature Modelling
Maintainability quality factor	Stability	X	X	X	X	X		X		X	X		X	X				
	Analyzability	X	X	X	X	X	X	X			X	X						
	Changeability	X	X	X	X	X	X	X				X		X	X	X	X	X
	Testability	X	X	X	X	X			X		X		X					

2.1.5 Proposing a Set of Hypotheses for Linking Subcharacteristics to Quality Attributes

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, Maintainability is described as a set of quality attributes. These quality attributes could be affected by one of various subcharacteristics as shown in Table 1. A first hypotheses is proposed in the table for the influences between subcharacteristics and quality attributes and they are showed for Maturity, MDE and Web Modelling characteristic and the Maintainability quality factor. This matrix could be made by expert designers and users and it has to be validated. The fuzzy logic is currently being considered in order to achieve real values, objectives and agreed.

2.2 The Other Components of the Framework

Other important element for QuEF is the *Thesaurus & Glossary* component. A thesaurus is a list containing the "terms" used to represent concepts, themes or contents of documents in order to make a terminological standardization to improve the access channel and communication between users of different MDWE methodologies.

We consider it necessary to carry out a standardization of terminology to improve the access channel for communication on MDWE. A set of concepts for MDWE methodologies is currently being described and related.

The Templates in the *Approach Characteristic Template component* with subcharacteristics and metrics for each characteristic are based on the Quality Model which is used to describe an input methodology. These templates would be used as input to QuEF. They would be analyzed in the evaluation process and compared with the model quality of the Quality Model component. Templates for MDE, Web Modelling, Tool Support, and Maturity have already been developed. In this work, Maintainability is studied in terms MDE, Web Modelling, and Maturity of the NDT methodology.

Finally, the *Quality Evaluation Process component* contrasts the information from each input approach template with information from the Quality Model. The main evaluation purpose is to identify tradeoffs and sensitivity points of the methodology under study. The idea is to determine which aspect needs to be improved on MDWE methodology. A simple evaluation is made using MS Excel which considers weights for metrics, subcharacteristics and quality attributes, although fuzzy logic is currently being considered in order to improve the evaluation process.

3 An Application of QuEF to the NDT Methodology

DT (Navigational Development Techniques) is an approach focused on the Model-Driven paradigm. Initially, NDT was focused on the requirements and analysis phases and it only defined a set of metamodels for the requirements and analysis phases and a set of transformation rules that let derive analysis models from requirements ones. Nowadays, it covers the complete life cycle of software development. In [8], some details about this evolution can be found. This evolution, some set of profiles let the use of UML tool based. Specifically, NDT-Suite uses these profiles to adapt Enterprise Architect [7] to use NDT. Besides, using the power of fusion and conexion of metamodels, NDT was extended to be adapted into enterprise environment. Nowadays, NDT is being used in several real projects and it is being involved in several new aspects like early testing or software quality.

3.1 Applying Templates to the NDT Methodology for the MDE, Web Modelling, and Maturity Characteristics

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 the Tool Support characteristic, MDE characteristic, Web Modelling, and Maturity characteristics can be considered. In this example, the Transformations subcharacteristic is shown in Table 2 as an example of a template of a subcharacteristic and metrics. Qualitative metrics indicate

Table 2 Transformation subcharacteristic and metric

Model-Driven Reverse Engineering or Synchronization			
This uses standard languages for defining synchronization methods or reverse engineering techniques such as ADM, XMI, MOF; GXL, JMI, EMF, MDR, QVT.	NS		
It supports a Reverse Engineering Tool: A tool intended to transform particular legacy or information artifact portfolios into fully fledged models.			
Model 2 Model	NS		
Code 2 Model	NS		
It provides a synchronization method or a reverse engineering technique between transformations such as:			
PIM2CIM	S	Code2CIM	NS
PSM2PIM	NS	Code2PIM	NS
PSM2CIM	NS	Code2PSM	S

if the subcharacteristic is Supported (S), Partly Supported (PS) or Not Supported (NS). The total value for the quality attribute is the number of values divided by the total metrics in the subcharacteristic. The metric value in the example is 1 if it is *supported,* 1/2 of the arithmetic mean of supported elements from among the total elements (for example in transformations) if it is *partly supported,* and 0 if it is *not supported*. When these metrics are quantitative, an average value is taken, while an expected value is set as an ideal value for comparison with the values in the results. Although none of the subcharacteristics of the MDE, the Web Modelling, and the Maturity is shown, they are considered in the evaluation process of this example in the following section.

For example, total metric values of MDE subcharacteristics are shown in Fig. 2. In the figure, black bars represent NDT metric values for each subcharacteristic of the MDE characteristic respectively and grey bars represent the expected values for an ideal approach. In the figure is seen that the Standard definition, Model-Based Testing and Transformation subcharacteristics may be improved for the NDT methodology. On the other hand, the NDT methodology has a good score in Levels of Abstraction and Traces. It means that new trends or other subcharacteristics have to be added in templates in order to perform a quality continuous improvement of quality.

3.2 An Evaluation of the Maintainability on NDT Methodology

In the implementation in Microsoft Excel, Functionality, Reliability, Portability, Usability and Maintainability quality factors have been studied. In this example, quality attributes of Maintainability are shown together with their relations with the MDE, Web Modelling, and Maturity characteristics. This is shown in Fig. 3,

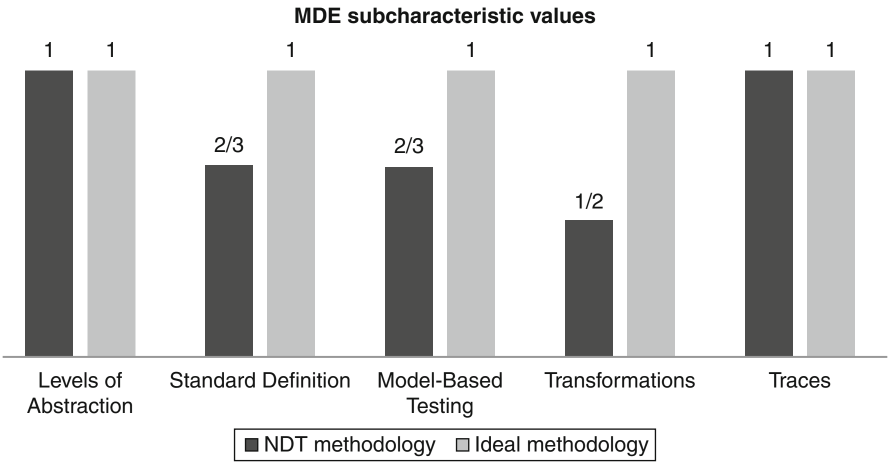

Fig. 2 Graph which represent maturity subcharacteristic values for NDT methodology

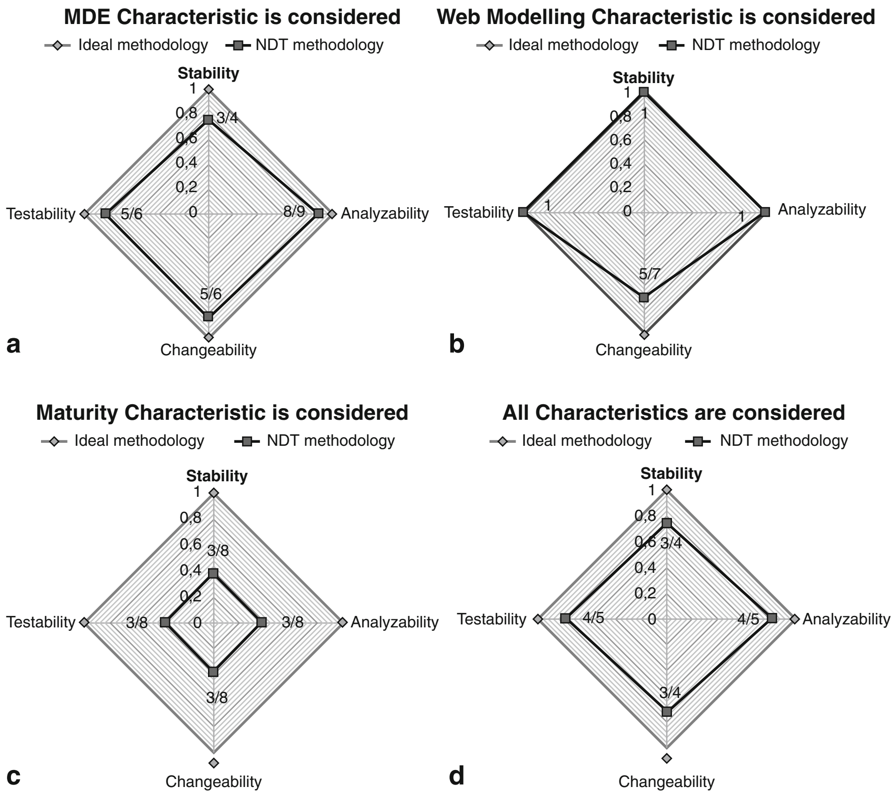

Fig. 3 Influences of several characteristics in the maintainability quality factor of the NDT methodology

where the black line represents Maintainability on the NDT methodology and the grey line represents the ideal Maintainability in an ideal approach, depending on the subcharacteristics under consideration. According to the results of the evaluation of the NDT methodology, only the MDE characteristic is considered in Diagram A, only Web Modelling is considered in Diagram B, and only Maturity is considered in Diagram C, for the evaluation of Maintainability. If all three characteristics are considered simultaneously then the results can differ greatly, as shown in Diagram D.

For the MDE characteristic, NDT has to improve the Stability quality attribute, which means that it would be difficult to avoid unexpected effects from modifications in the approach. For the Web Modelling characteristic, NDT has to improve the Changeability although it yields good results in Stability, Testability and Analyzability. In this graph we can see that results are uniform for this set of quality attributes (Testability, Stability and Analyzability), which could be due to: the similarity in the results; or the necessity to have more subcharacteristics and metrics for the identification of differences between these quality attributes, whereby characteristic templates would have to be defined in greater descriptive detail. In the Maturity graph, it can be observed that Maturity improves and renders the results more uniform. In general terms where all characteristics are considered, (Diagram D) NDT yields good results in Analyzability and Testability but has to improve the Stability and Changeability.

4 Conclusions

A framework is needed for the improvement of current proposals, and would be highly useful for the successful development of a new MDWE methodology. Therefore a quality environment for the assessment of MDWE methodologies is proposed. We consider that the use of QuEF will enhance the quality of products, processes and techniques of approaches. Furthermore, QuEF could be used for the optimization of a continuous improvement in quality since the number of sub-characteristics selected can be reduced, by using the matrix of influences, to include only those with the major influence in quality attributes. In previous papers [5, 6], we evaluated subcharacteristics related with MDE, Maturity and Tool Support of the NDT Methodology which are required for the measurement of the value of MDWE methodologies so that they can be assessed and improved in terms of their Functionality, Usability, Portability or Reliability. Therefore the use of QuEF can improve the efficiency and effectiveness of MDWE methodologies and in turn may lead to their more widespread use since this evaluation approach helps one understand the strengths and weaknesses of a methodology. This environment could be extended to involve other areas or domains. Further characteristics and quality factors have yet to be developed. To this end, Microsoft Excel is employed as a first prototype, although a software tool remains to be developed. Other methodologies, such as WebML and UWE [12, 16] are currently being evaluated in order to discover how they may be improved.

Acknowledgements This research has been supported by the project QSimTest (TIN2007–67843-C06_03) and by the Tempros project of the Ministry of Education and Science (TIN2010–20057-C03–02), Spain.

References

1. Abrahão S, Condori-Fernàndez N, Olsina L, Pastor O (2003) Defining and validating metrics for navigational models. IEEE computer society. In: Proceedings of the 9th international software metrics symposium (METRICS'03), Australia, p 200
2. Briand LC, Melo WL, Wüst J (2006) Assessing the applicability of fault-proneness models across object-oriented software projects. ISERN Report No. ISERN-00-06, Version 2, pp 1–20
3. Cachero C, Calero C, Marhuenda Y (2008) A quality-aware engineering process for web applications. In: Handbook of research on web information systems quality. Information Science Publishing, Hershey
4. Calero C, Ruiz J, Piattini M (2005) Classifying web metrics using the web quality model. Online Inf Rev 29(3):227–248
5. Domínguez-Mayo FJ, Escalona MJ, Mejías M, Ramos I, Fernàndez L (2010) A quality evaluation framework for MDWE methodologies. In: Proceedings of the 18th international conference on software quality management (SQM 2010), vol 1, No. 1, pp 171–184
6. Domínguez-Mayo FJ, Escalona MJ, Mejías M, Torres AH (2010) Towards a quality evaluation framework for model-driven web engineering methodologies. 6th international conference on web information systems and technologies (WEBIST 2010), vol 2, pp 191–194
7. Enterprise Architect. http://www.sparxsystems.com/
8. Escalona MJ, Aragón G (2008) NDT. A model-driven approach for web requirements. IEEE Trans Softw Eng 34(3):377–390
9. Escalona MJ, Koch N (2004) Requirements engineering for web applications—a comparative study. J Web Eng 2(3):193–212
10. García F, Serrano M, Cruz-Lemus J, Ruiz F, Piattini M (2007) Managing software process measurement: a metamodel-based approach. Inf Sci 177(12):2570–2586
11. ISO—International Organization for Standardization, ISO/IEC 9126–1. http://www.iso.org
12. Kroiß C, Koch N (2008) 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
13. Lee Y-S, Liang B-S, Wu S-F, Wang F-J (1995) Measuring the coupling and cohesion of an object-oriented program based on information flow. In: Proceedings of the international conference on software quality. Maribor, Slovenia
14. Mohagheghi P, Aagedal J (2007) Evaluating quality in model-driven engineering. In: International workshop on modeling in software engineering (MISE'07). IEEE Computer Society, Washington, DC
15. OMG: MDA. http://www.omg.org/mda/faq_mda.htm
16. Schwinger W, Retschitzegger W, Schauerhuber A, Kappel G, Wimmer M, Pröll B, Cachero Castro C, Casteleyn S, De Troyer O, Fraternali P, Garrigos I, Garzotto F, Ginige A, Houben G-J, Koch N, Moreno N, Pastor O, Paolini P, Pelechano Ferragud V, Rossi G, Schwabe D, Tisi M, Vallecillo A, van der Sluijs K, Zhang G (2008) A survey on web modeling approaches for ubiquitous web applications. Int J Web Inf Syst 4(3):234–305