

Quality evaluation for Model-Driven Web Engineering methodologies

F.J. Domínguez-Mayo^a, M.J. Escalona^a, M. Mejías^a, M. Ross^{b,c}, G. Staples^c

^a Department of Computer Languages and System, University of Seville, Seville, Spain

^b Faculty of Maritime and Technology, Southampton Solent University, Southampton, United Kingdom

^c BCS Software Quality Specialist Group, United Kingdom

Keywords:
Software engineering
Standards
Metrics/measurement
Methodologies
Quality analysis and evaluation
Software quality/SQA

A B S T R A C T

Context: There are lots of approaches or methodologies in the Model-Driven Web Engineering (MDWE) context to develop Web Applications without reaching a consensus on the use of standards and scarcity of both, practical experience and tool support.

Objective: Model-Driven Web Engineering (MDWE) methodologies are constantly evolving. Moreover, Quality is a very important factor to identify within a methodology as it defines processes, techniques and artifacts to develop Web Applications. For this reason, when analyzing a methodology, it is not only necessary to evaluate quality, but also to find out how to improve it. The main goal of this paper is to develop a set of Quality Characteristics and Sub-Characteristics for MDWE approaches based on ISO/IEC standards.

Method: From the software products context, some widely standards proposed, such as ISO/IEC 9126 or ISO/IEC 25000, suggest a Quality Model for software products, although up to now, there are no standard methods to assess quality on MDWE methodologies. Such methodologies can be organized into Properties, thus, a methodology has artifacts, processes and techniques. Then, each item is evaluated through a set of appropriate Quality Characteristics, depending on its nature. This paper proposes to evaluate a methodology as a product itself.

Results: This paper recommends a set of Quality Characteristics and Sub-Characteristics based on these standards in order to evaluate MDWE methodologies quality. Additionally, it defines an agile way to relate these Quality Sub-Characteristics to Properties with the sole purpose of not only analyzing, but also assessing and improving MDWE methodologies.

Conclusions: The application of these Quality Characteristics and Sub-Characteristics could promote efficiency in methodologies since this kind of assessment enhances both the understanding of strengths and weaknesses of approaches.

1. Introduction

Quality is a relevant aspect to consider in the Software Engineering context, although there are several different definitions, for example, conformance to user expectations, which is often described as the “fitness for purpose” of a piece of software. Another quality definition attending to software quality measures deals with the high quality of software design (quality of design) and the high level the software conforms with that design (quality of conformance). The definition of quality in ISO 9000:2005, as described in [21], includes the notion of “degree” meaning that quality is not an absolute, but a changeable aspect. The concept of “degree” is illustrated in Fig. 1, showing that needs, requirements and expectations are constantly changing. Therefore, quality is

the difference between the model state implied or required, and the model state reached. Thus, satisfactory quality takes place when the state reached is within the range of acceptability defined by the required model; superior quality occurs when the state reached is above the required model and inferior quality appears when the state reached is below the required state.

In the final evaluation, it is the user who sets the quality standards by deciding which products should be purchased and whom they should be purchased. We need to express our relative satisfaction with products and, as a consequence, use subjective terms. When a product satisfies the user’s needs, it can be said that the product is either a high quality or a satisfactory quality product and in the same way, when the user is dissatisfied with the product, it can be said that it is either, a poor or a low quality product. When the product exceeds the user’s needs, it is considered to be either a high quality or a superior quality product, whereas if it is below the user’s expectations, it is considered to be either a low quality or unsatisfactory quality product.

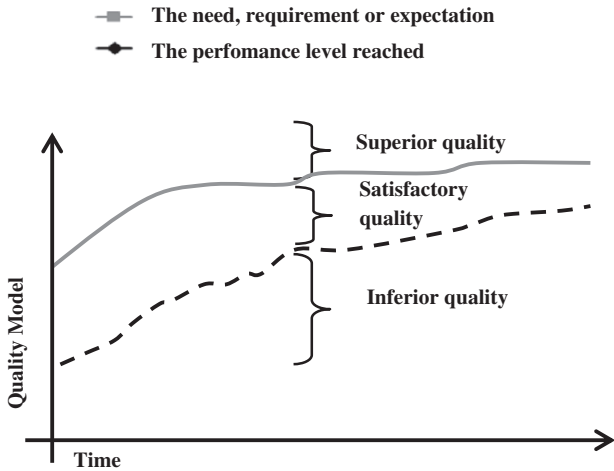


Fig. 1. Quality in ISO/IEC.

The Model-Driven Engineering (MDE) is a software development paradigm dealing with the creation of models or abstractions closer to a particular domain than to concepts or specific syntax. A Web Engineering domain specific to MDE is called MDWE (Model-Driven Web Engineering). A Model-Driven Architecture (MDA) [37] is an architecture platform approach to develop software systems under the MDE paradigm which provides a set of guidelines to structure specifications that are expressed as models.

In recent years, the growing interest in the Internet has generated a high number of MDWE approaches [45] that offer a frame of reference for Web environment. Nowadays, there are many approaches and diverse MDWE methodologies such as OOHD (Oriented Hypermedia Design Method) [40], UWE (UML-based Web Engineering) [44], WebML (The Web Modeling Language) [48], OOH4RIA [34], RUX-Method [39] or NDT (Navigational Development Techniques) [36], which do not reach a consensus on the use of standards, on the one hand, and show scarcity of both practical experience and tool support, on the other. Thus, each methodology has different tools, such as metamodels (some of them are based on other methodologies) or transformations, which can implement different levels of abstractions, for instance, the Computer Independent Model (CIM), the Platform Independent Model (PIM) and/or the Platform Specific Model (PSM), among others. In this situation, it is necessary to characterize these methodologies in order to analyze and evaluate them.

In addition, it is important to know both, the real needs that designers have to cover regarding users of these approaches and also the Quality Characteristics these needs should guarantee. Thus, designing a clear strategy is essential since it will allow designers to efficiently outline these methodologies. QuEF [13] is a framework which manages quality in MDWE approaches. It is based on quality management, but it focuses on the Quality Model lifecycle. This lifecycle is composed of a set of phases (Strategy phase, Design phase, Transition phase, Operation phase and Quality Continuous Improvement phase) that helps quality management work effectively. In QuEF, the Quality Model is the key element in quality management since it describes all the necessary elements that make the automatic generation of artifacts reduce the estimated time and cost.

The Strategy and Design phase must start with defining all users' needs in general. Once all these needs are customized then, in turn, they are defined with a Quality Model. During the Operation phase, the most appropriate one is selected and both users and designers can analyze, control and evaluate the quality of their approaches, as shown in the sample application included in this

paper. Users of methodologies need to find out the most appropriate one for themselves and their work environment. Due to their experience, they would prefer some aspects rather than others

In any case, users have the last word to decide and designers must offer only what users need.

All metrics studied in this paper focus on determining the aspects that must be included in a methodology. Thus, in the Operation phase, users select their preferences by customizing a set of weight values associated to the Quality Characteristics analyzed in this paper as well as the Properties containing users' needs and environmental description of approaches in terms of the importance given.

This paper focuses on developing a set of Quality Characteristics and Sub-Characteristics for MDWE approaches based on ISO/IEC standards. It aims to propose the bases of a set of these Quality Characteristics as part of the Strategy and Design phase of QuEF. We also suggest how to analyze, control and evaluate the quality of MDWE approaches as part of the Operation phase. Besides, this paper deals with defining Quality Characteristics. The idea is to provide an environment that allows users and designers to figure out which of the Quality Characteristics have to be guaranteed when particular users apply these methodologies.

The paper is organized into the following sections: after this introduction, Section 2 presents a global analysis of the situation and all necessary elements to elaborate this work according to some quality standards and contexts. Section 3 proposes quality concepts such as Quality Characteristic and Quality Sub-Characteristic as part of the Strategy and Design phase in QuEF. Section 4 explains how all these concepts are related to a Matrix of Influences (Mols) as well as how it provides a set of formulas to analyze and evaluate MDWE methodologies as part of the Operation phase in QuEF. In Section 5, a set of Quality Characteristics and Quality Sub-Characteristics are identified. Section 6 provides a sample of the proposed analysis and NDT methodology assessment. Finally, Section 7 and 8 offer a set of conclusions and contributions and suggest possible future work.

2. Work context and related work

Few years ago, several research groups began to analyze the characteristics of new emerging software systems known as hypermedia systems, which have eventually evolved into Web systems. It was the birth of a new line of Software Engineering currently known as Web Engineering [16]. It is a specific domain within MDE (Model-Driven Engineering) paradigm where an application can be used [15]. The application of MDE Engineering to Web sites is called Model-Driven Web Engineering (MDWE) and, as it can be observed in different studies [7,45,15], it is offering very good results. Nowadays, there are several proposals on MDWE in the literature that are very useful for designing such applications. Some of them almost entirely cover every level of abstraction and they even have tools that support transformations automation in development and evaluation processes.

There is a variety of proposals in Web Engineering, as shown in [45]. This range of possibilities and the trend towards using MDE in proposals, open such a wide range of offers that, in many cases, it is difficult to select the most appropriate one. MDE was launched by the Object Management Group (OMG). The OMG has also developed the proposed MDA (Model-Driven Architecture) that provides standard platform architecture for proposals based on the Model-Driven paradigm. MDA was created with the idea of separating a system logical specification from the operational details that define how the system uses the technology platform capabilities to be implemented. In this regard, the goals of MDA are portability, interoperability and reusability through architectural separation.

Besides, the concept of platform independence frequently appears in MDA, which models the independence of the characteristics of any technological platform in CIM and PIM level of Abstraction.

Among the benefits provided by MDE, it can automate the software development, documentation, code generation, testing and project management. This fact favors a growing productivity in software development and maintenance, as well as time and cost reduction. Finally, it also increases the quality of the products obtained. By applying this paradigm, the lifecycle of a software system is completely covered, ranging from requirements capture to maintenance, or code generation.

There are several methodological approaches and many comparative studies in the MDWE area. In this sense, Schwinger et al. [45] must be taken into account. The authors specifically consider modeling concepts in their ubiquitous nature. They also include a study regarding the available support for Model-Driven Development in a comprehensive way, using a well-defined and fine-grained catalogue with more than 30 evaluation criteria.

As far as quality is concerned, Heitlager et al. [19] discuss that the amount of effort needed to maintain a software system is related to the technical quality of the source code of that system. Besides, the ISO/IEC 9126 does not provide a consensual set of measures for estimating maintainability on the basis of the source code of a system. Heitlager et al. [19] identify a number of requirements to be fulfilled by a maintainability model in order to be usable in practice. They also discuss their experiences by using such a system for IT management consultancy activities. Azuma [3], on the contrary, introduces an overview of current ISO/IEC 9126, SQuaRE and ISO/IEC 14598 series, followed by some problems to be solved. Besides, Azuma explains new parts, such as Quality in Use, Quality Requirements and Elementally Metrics. Herrera et al. [20], on their part, aim at defining a Quality Model to assess Web Portals quality. The model is based on ISO/IEC 25010 standard and some related work found in the literature. Finally, Calero et al. [7] present the WQM (Web Quality Model), which distinguishes three dimensions related to web features, lifecycle processes and Quality Characteristics, and classify the most relevant web metrics through the framework.

The idea of developing a MDE framework for evaluating quality has been applied in several studies by Mohagheghi and Dehlen [35], where they state that the quality of models is affected by the quality of modeling languages, tools, modeling processes, the knowledge and experience of modelers and the quality assurance techniques applied. Moreover, there are different proposals concerning conceptual frameworks for the metrics and indicators domains, such as different metamodels or tools like GenMETRIC [17], a proposal for the integrated management of software measurement. GenMETRIC aims at providing companies with a generic and flexible environment for software measurement, by facilitating and setting the bases for a common and effective measurement process, not only restricted to one kind of software entity, a single quality or an evaluation mode. Another framework is C-INCAMI [6], a goal-oriented and organization Measurement and Evaluation Framework. C-INCAMI framework creates an ontological domain model defining all the concepts and relations needed to design and implement measurement and evaluation processes. It is an approach in which the requirements specification, measurement and evaluation design, and analysis of results are elaborated to satisfy specific information in a given context. This ontology also considers sources in terms of WebQEM terminology such as ISO/IEC standards, articles and books.

Olsina and Rossi [39] explain the use of WebQEM, a quantitative evaluation strategy to assess web sites and the quality of applications. Defining and measuring quality indicators can help stakeholders understand and improve web products. An e-commerce case study illustrates the use of methodology in systematically

assessing attributes that influence the quality of the product. Olsina et al. [38] describe more than a hundred characteristics and attributes for the academic sites domain in order to analyze the quality requirement tree and the way to specify them. These elements are used in a quantitative methodology for assessment, comparison and ranking processes. The proposed web site Quality Evaluation Methodology (QEM) can be useful to assess quality in different phases of a web product lifecycle.

Punter et al. [43] suggest the W-process. It supports the selection and assessment of appropriate techniques in an evaluation, for example, by choosing the right type of metrics to measure software product quality. The W-process is an extension of ISO/IEC 14598, taking the activities of the standard into account, but expressing the relations among them more explicitly than the standard does. The process carries out a categorization of the ISO/IEC 14598 activities in three levels, namely: Goal, Question and Metric. In this line, the open literature typically describes GQM (Goal Question Metric) [5] in terms of a six-step process where the first three steps deal with using business goals to drive the identification of the right metrics and the last three steps deal with gathering measurement data and making effective use of measurement results to drive decision-making and improvements.

Plösch et al. [41] present a method to systematically collect quality requirements for software products based on quality models and potential obstacles that may obstruct quality goals.

In previous work [11,10], a framework to analyze, evaluate and improve quality of MDWE approaches is posed but this version of QuEF was actually extended in [13] with several phases: Strategy phase, Design phase, Transition phase, Operation phase and Quality Continual Improvement phase. Such phases have been defined together with processes and artifacts to cope with a complete Quality Model lifecycle. In QuEF extension the main difference among frameworks is that quality management is focused on the Quality Model lifecycle. The purpose of QuEF extension is to converge on a continuous automatic quality improvement by means of generating Checklists and documentation as well as automatic evaluations and plans in order to control and better quality and, in turn, automatically reduce effort and time.

On the one hand, this paper points out some relevant details when defining a set of Quality Characteristics and Quality Sub-Characteristics for the quality assessment of MDWE approaches as a part of the Strategy and Design phase of QuEF and, on the other, it also proposes a way to analyze them as a part of the Operation phase.

As far as Quality standards are concerned, ISO published [1] an expanded version, containing ISO quality models and a consensus decision on inventories of proposed measures for these models. The current version of ISO/IEC 9126 series of standards consists of four documents:

- ISO/IEC 9126-1: Quality Models [25].
- ISO/IEC TR 9126-2: External Metrics [26].
- ISO/IEC TR 9126-3: Internal Metrics [27].
- ISO/IEC TR 9126-4: Quality in Use Metrics [28].

As a result of Information Technology advances and environmental changes, a new version of ISO/IEC 9126 has recently appeared, known as ISO/IEC 25000:2005 [29]. It guides on the use of the new series of International Standards named Software product Quality Requirements and Evaluation (SQuaRE). The purpose of this guidance is to provide an overview on SQuaRE contents, common reference models and definitions, and relations among documents. This makes users properly understand these series of International Standards according to their purposes. The document explains the transition process from the old ISO/IEC 9126 and ISO/IEC 14598 [31] series to SQuaRE, and it also gives information

about using ISO/IEC 9126 and ISO/IEC 14598 series in their previous form. The aim is to guide the software product development with quality requirements specification and evaluation. The document set criteria for quality requirements specification in software products, their metrics and evaluation.

SQuARE consists of the following divisions [30]:

- ISO/IEC 2500n: Quality Management Division.
- ISO/IEC 2501n: Quality Model Division.
- ISO/IEC 2502n: Quality Measurement Division.
- ISO/IEC 2503n: Quality Requirements Division.
- ISO/IEC 2504n: Quality Evaluation Division.

One of the objectives of ISO/IEC 25000 series (and what makes them different from the current ISO/IEC 9126 series) is the harmonization of its contents with the software measurement terminology of ISO/IEC 15939:2007, which determines a measurement process applicable to System and Software Engineering and management disciplines alike. The process is described through a model that defines the necessary measurement process activities to adequately specify what measurement information is required, how the measures and analysis results are applied and how to consider their validity. The measurement process is flexible, tailorable and adaptable to different users' needs. ISO/IEC 15939:2007 identifies a process that enhances defining a suitable set of measures addressing specific Properties.

Other rigorous and disciplined methodology is Six sigma [21] which uses data and statistical analysis to measure and improve a company's operational performance by identifying and eliminating process "defects". In this line, Software Quality Assurance (SQA) is responsible for monitoring Software Engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may ensure conformance to one or more standards, such as ISO 9000 or a model such as CMMI. SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, management change, management configuration, testing, management release and product integration. SQA is organized into goals, commitments, abilities, activities, measurements and verifications. In this context, ISO/IEC 9126 is a standard Quality Characteristic terminology for software product evaluation.

ISO/IEC 15504 [32], also known as SPICE (Software Process Improvement and Capability determination), consists of a set of technical standard documents for the computer software development process and the related business management functions. It is the reference model for the maturity models (dealing with capability levels which in turn have to do with the process attributes and generic practices) against which the assessors can place the evidence collected during their assessment, so that, they can give an overall idea of the organization's abilities for delivering products (software, systems, and IT services). In this line, ISO/IEC 12207:2008 [33] establishes a common framework for software lifecycle processes, with well-defined terminology, that can be referenced by the software industry. ISO/IEC 12207:2008 also provides a process that can be used for defining, controlling and improving software lifecycle processes. Processes, activities and tasks of ISO/IEC 12207:2008 - either alone or in conjunction with ISO/IEC 15288 - may also be applied during the acquisition of a system containing software.

ISO/IEC 14598 series of standards provide methods for measuring, assessing and evaluating software products quality. They describe neither methods for evaluating software production processes nor methods for cost prediction (software product quality measurements may, of course, be used for both purposes) [21]. ISO/IEC 19796-1:2005 is an interesting standard to consider when developing a framework for the quality evaluation of MDWE

methodologies. This standard is focused on e-learning and aims to harmonize the different approaches used around the world for assessing e-learning quality initiatives. The e-learning market acceptance depends on the quality of the related products, services and tools. A harmonized conception of e-learning quality is a prerequisite for a properly functioning market on e-learning products and services and their constant quality improvement.[23]

3. A quality management framework based on the Quality Model Lifecycle

Given the high number of methodologies available and recently proposed, it has become necessary to define objective management tools to enable organizations to improve their methodological environment and help web methodologies designers create new effective and efficient tools, processes and techniques. Since methodologies are constantly evolving, it is not only necessary to manage and improve quality but also to optimize the process in order to reduce costs and time. QuEF [11,10] is a quality management framework for Model-Driven Web methodologies. In the present work, a Model-Driven Web methodology is an approach for Web Applications development based on the MDE paradigm that provides a set of guidelines, techniques, processes and/or tools to structure specifications which are expressed as models.

QuEF was initially developed for the analysis and evaluation of these kinds of methodologies and it has recently been extended to cover different phases of the quality management. In Domínguez-Mayo et al. [13], all the phases have been described together with the objectives and artifacts they used. These phases, as shown in Fig. 2, have been identified in a lifecycle focusing on the Quality Model.

- **Quality Model Strategy phase:** This is the initial phase and the main objective concludes that quality management becomes strategically active. In this phase all the strategic assets have to be identified.
- **Quality Model Design phase:** In this phase the Quality Model has to be designed in terms of all strategic views resulting from the previous phase. Significantly, a design within QuEF is understood to encompass all relevant elements that are to be designed within the Quality Model. This Quality Model will be used in the Operation phase for the quality management.
- **Quality Model Transition phase:** The Quality Model transition phase includes changing the Quality Model, without influencing the operation phase. This phase is very important and it needs to keep pace with changes in technology, processes or techniques. These changes do not have to influence quality management but do have to be considered in the Operation phase.
- **Quality Model Operation phase:** In this phase the Quality Model is used to manage the quality of approaches. All artifacts like Checklists, evaluations or plans, among others, are automatically generated in terms of the Quality Model and designed in the Design phase. The analysis and evaluation of quality in the Operation phase is carried out to control and evaluate the current state of quality. Finally, all results are used for the continuous improvement of quality in the Continual Improvement phase.
- **Quality Continual Improvement phase:** The Quality Model can change due to changes of trends or technologies. In addition, this phase aims to align and realign the Quality Model with the Properties to be covered and Quality Characteristics to be assured along with the improvement of every phase in the lifecycle.

Basically, this way to manage quality consists of defining a Quality Model containing all past, present and future require-

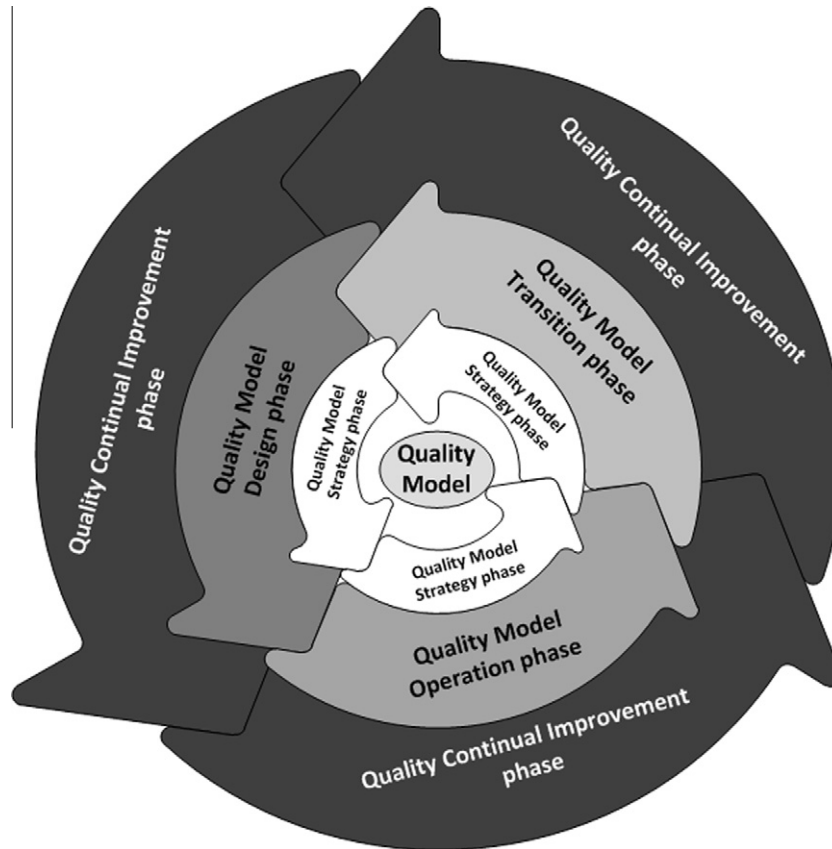


Fig. 2. Quality model lifecycle in QuEF framework.

ments, expectations, goals or trends to reach. The idea is to centralize all efforts and costs concerning quality management in a unique model. Therefore, the effort is reduced not only by centralizing the model but also by generating all necessary artifacts like Checklists, Improvement plans or Charts, among others, by means of a Model-Driven Quality.

QuEF can be applied to any domain that may define goals and requirements to reach. The benefits or ROI are costs and time reduction when improving the domain under study. Besides, knowledge can be shared by stakeholders of the same domain. Another advantage lies in our strategic map for managing quality efficiently. Despite the initial effort to implement QuEF, if the defined strategy is clear and well defined, effort is a good investment since it increases quality control, faster quality improvement and competitiveness.

The lifecycle deals with the set of phases performed to carry out quality management. In this lifecycle, past, present and future views of the Quality Model elements in the domain under study must be considered in order to get quality management in an efficient way. This phase represents a strategic asset that focuses on defining a quality management strategy. Moreover, the Quality Model Strategy phase of QuEF is the key issue in quality management. The Quality Model is finally designed in the QuEF Quality Model Design phase, where all strategic activities from the previous phase are taken into consideration. In fact, this Quality Model will be the one used for Quality analysis, control and evaluation in the next Operation phase.

The Quality Model Operation phase is used to carry out the Quality management. Thus, the Analysis and evaluation management processes are performed within this phase which means that we can know the current state of the domain under study. If the

domain or context changes due to the appearance of new trends, then the Quality Model needs to be changed in the Quality Model Transition phase without affecting the Operation phase. Finally, the Quality Continual Improvement phase executes all processes to improve quality in all the lifecycle processes as well as in the Quality Model itself.

The use of QuEF focuses on medium-long term quality management. Additionally, a short-term use of QuEF for Quality management is also possible, but considering some aspects. Particularly, in a short-term quality management, Checklists are used to assure quality (meeting the Quality Model, for example, in terms of requirements or objectives, among others) rather than to analyze the current state domain under study. On the contrary, if a medium-long term Quality management is considered, then, the Checklists determine the current state rather than ensuring compliance with the Quality Model. In this sense, despite performing a short or a medium-long term quality management, introducing QuEF always requires a great initial effort rewarded with the aforementioned benefits.

In the Strategy phase of QuEF, some of the main necessary processes to be carried out are, among others, the Property Portfolio Management, the Quality Characteristic Portfolio Management and the Quality Model Portfolio Management. Different portfolios are obtained as a result of these processes:

- The Portfolio of Properties.
- The Portfolio of Quality Characteristics.
- The Portfolio of Quality Model.

These portfolios are used as strategic assets to obtain the final Properties, the Quality Characteristics and the Quality Model in

the Design phase. When the Portfolios are active and have past, present and future information, the Design phase uses such information to outline the models.

Both Properties and Quality Characteristics have to be described and structured in a two-levelled hierarchy which represents a higher and lower level of granularity of an approach. Properties are used to describe and categorize the proposals in two levels (Feature and Sub-Feature) and they indicate the degree to which a methodology is implemented. For instance, the MDE Feature may have some Sub-Features such as, Language Definition, Transformations and Trace Generation.

On the contrary, Quality Characteristics and Quality Sub-Characteristics represent the topic of this work. They need to be clarified, but they are quality aspects influenced by an environment description or Properties. In other words, a Quality Characteristic is a higher-level quality aspect. In a hierarchy of Quality Characteristics, higher-level attributes are called Quality Characteristics and lower-level attributes are called Quality Sub-Characteristics. For example, Quality Characteristics could be Usability, Functionality or Portability. Usability, for instance, is defined by different Quality Sub-Characteristics such as Learnability, Understandability or Operability. Fig. 3 shows the relation among the aspects highlighted in this paper. A Quality Characteristic has a set of Quality Sub-Characteristics and a Feature has a set of Sub-Features.

3.1. A Strategy phase based on a Systematic Literature Review

In QuEF, the Technological Watch Management is a process of the Quality Model Strategy phase focused on the observation of trends and technologies on the rise. It also explores the state of the art of the domain. Due to the definition of Properties (where CheckLists are defined) and Quality Characteristics involves a large initial effort this task is made by carrying out a Systematic Literature Review (SLR).

There are many reasons for undertaking a SLR such as the identification of concepts and gaps for MDWE methodologies quality management. The results of this process are used to capture information for Properties and Quality Characteristics Portfolios.

The review method is based on the research protocol and it is in this section where sources, studies selection and selection execution are defined.

- **Sources Selection:** The objective of this section is to select the sources where searches for primary studies will be executed. First of all, we carried out preliminary searches aimed at identifying existing systematic reviews and assessing the vol-

ume of potentially relevant studies. In this sense, we found the following relevant initiatives: Schwinger et al., 2008, Meliá et al., 2008, Preciado et al., 2008, Escalona and Koch 2004 or Escalona and Aragón, 2008, among others.

The proposed study by Schwinger et al. was a good basis for this framework, although it is not sufficient because the environment does not only aim at assessing approaches, but managing quality as a whole. In addition, in a good strategy, the Properties that are shown in the study are not sufficient since they must be considered in a quality continuous improvement regarding past, present and future trends. This favors a good quality design and leads to a quality management based on quality continuous improvement.

After that, a Web-search was performed to find other relevant and new concepts related to our Properties and Quality Characteristics. Several sets of keywords were used by combining the concepts of our study. Some examples are: “Model-Driven”, “Web”, “development”, “methodology”, “methodologies”, “approach”, “approaches”, “CIM”, “PIM”, “SPM”, “Computer Independent Model”, “Platform Independent Model”, “Specific Platform Model”, Code”, “Level of abstraction”, “Tool Support”, “evaluation”, “assessment”, “Quality”, “ISO 9126”, “ISO 25000”, “CMMI” or “ITIL”, among others. The last step consisted in looking for references of papers included in the previous reviews in order to identify more concepts.

- **Studies selection:** Then the selection criteria to evaluate studies sources were: the availability to consult articles on the Internet or on the digital library of the University of Seville, which has e-books and also access to other resources like Google Scholar, Scopus, Mendeley, Science Direct, ISI Web of Knowledge, ACM Digital Library, CiteSeerX or the IEEE digital library; availability for search mechanism through keywords; and literature from companies, books, journals and conferences published by experts in the field.

As there are plenty of work published, our search started excluding those issued prior to 2006 because we were interested in current work. Then we analyzed which of them covered our domain, and all studies dealing with the Model-Driven Web development environments domain were included. The process to consider a paper was as follows: First, the title was considered, then the Abstract, Keywords and Content and finally the Conclusions of the analysis.

- **Selection execution:** Then, the procedures for studies selection were applied to obtain articles in order to verify if the studies fit the inclusion and exclusion criteria. Finally, attending to the

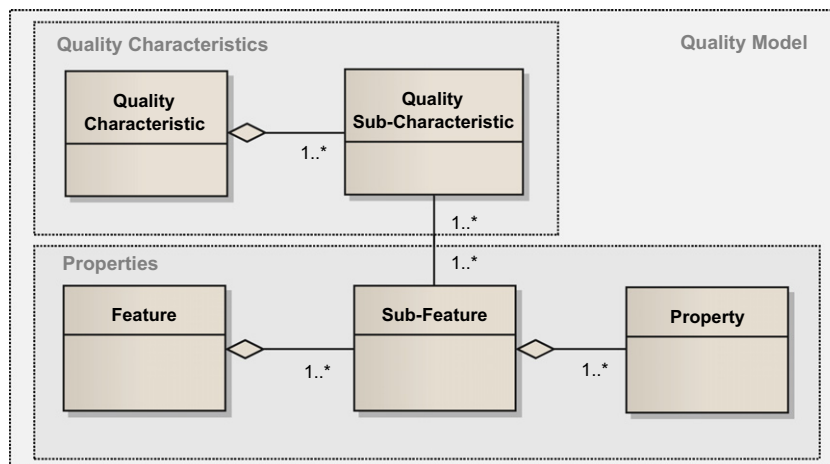


Fig. 3. Quality model metamodel.

work. The Quality Sub-Characteristic values $s_{q_{k,l_k}}$ are calculated according to formula (1), which considers each Sub-Feature value $r_{f_{i,j}}$ of a Feature F_i . This formula calculates the Quality Sub-Characteristic value in terms of those Sub-Features that influence the Quality Sub-Characteristic. This is defined by an association link that is a qualitative value in the Mol, which is, then, transformed into a quantitative value $v_{[f_{i,j},q_{k,l_k}]}$. Besides, the importance of each Sub-Feature in the set of Sub-Features is considered by the Sub-Feature weight value $w_{f_{i,j}}$.

$$s_{q_{k,l_k}} = \frac{\sum_{j=1}^n v_{[f_{i,j},q_{k,l_k}]} r_{f_{i,j}} w_{f_{i,j}}}{n} \quad (1)$$

- $s_{q_{k,l_k}}$ Z. It is a Quality Sub-Characteristic value. It is between $(0 \leq s_{q_{k,l_k}} \leq 1)$ because this value actually represents whether the state reached by this value satisfies or not the required model (value = 1).
- $r_{f_{i,j}}$ Z. It is a Sub-Feature value between $(0 \leq r_{f_{i,j}} \leq 1)$ since this value is actually a degree representing whether the state reached by this value satisfies or not the required model (value = 1).
- $v_{[f_{i,j},q_{k,l_k}]}$ Z. It is the degree of influence value or the relation value between the Sub-Feature $f_{i,j}$ and the Quality Sub-Characteristic q_{k,l_k} . In the Mol, it is a qualitative value that has to be changed into a quantitative value between $(0 \leq v_{[f_{i,j},q_{k,l_k}]} \leq 1)$. It depends on the value scale defined indicating the influence.
- $w_{f_{i,j}}$ Z. It is the weight value of the Sub-Feature $f_{i,j}$. It is between $(0 \leq w_{f_{i,j}} \leq 1)$ and represents the importance of the Sub-Feature in the set of Sub-Features. The sum of this value for the set of Sub-Features of the same Feature has to be 1.
- n Z. It is the number of Sub-Feature values of a Feature F_i whose degree of influence value $v_{[f_{i,j},q_{k,l_k}]}$ is not 0.
- Nevertheless, it can be generalized and the Quality Sub-Characteristic value $s_{q_{k,l_k}}$ can be calculated in terms of a Feature F_i or, on the contrary, considered as a combination of a set of Features. For instance, these two Features are considered F_1 and F_2 , as indicated in formula (2).

$$s_{q_{k,l_k}} = \frac{\sum_{j=1}^{n_1} v_{[f_{1,j_1},q_{k,l_k}]} r_{f_{1,j_1}} w_{f_{1,j_1}} + \sum_{j=1}^{n_2} v_{[f_{2,j_2},q_{k,l_k}]} r_{f_{2,j_2}} w_{f_{2,j_2}}}{n_1 + n_2} \quad (2)$$

In formula (2), n_1 and n_2 are the numbers of Sub-Feature values of each Feature F_1 and F_2 whose weight value $v_{[f_{1,j_1},q_{k,l_k}]}$ and $v_{[f_{2,j_2},q_{k,l_k}]}$ respectively, are not 0.

In other words, it can be generalized to this other formula (3) where a Quality Sub-Characteristic value $s_{q_{k,l_k}}$ is calculated in terms of different Features F_1, F_2, \dots, F_t selected.

$$s_{q_{k,l_k}} = \frac{\sum_{j=1}^{n_1} v_{[f_{1,j_1},q_{k,l_k}]} r_{f_{1,j_1}} w_{f_{1,j_1}} + \dots + \sum_{j=1}^{n_t} v_{[f_{t,j_t},q_{k,l_k}]} r_{f_{t,j_t}} w_{f_{t,j_t}}}{n_1 + \dots + n_t} \quad (3)$$

In a similar manner to formula 2, n_1, \dots, n_t represents the number of Sub-Feature values of each Feature F_1, \dots, F_t whose degree of influence values $v_{[f_{1,j_1},q_{k,l_k}]}, \dots, v_{[f_{t,j_t},q_{k,l_k}]}$ respectively, are not 0.

Finally, every Quality Characteristic value s_{Q_k} is calculated by considering each Quality Sub-Characteristic q_{k,l_k} and the necessary degree to influence it with a weight value. In the following formula (4), $w_{q_{k,l_k}}$ represents the weight value that indicates the influence of each Quality Sub-Characteristic value q_{k,l_k} on the Quality Characteristic value s_{Q_k} associated.

$$s_{Q_k} = \frac{\sum_{l=1}^m w_{q_{k,l_k}} s_{q_{k,l_k}}}{m} \quad (4)$$

- s_{Q_k} . It is a Quality Characteristic value. It is between $(0 \leq s_{Q_k} \leq 1)$ because this value actually represents a degree showing whether the state reached by this value satisfies or not the required model (value = 1).

- $w_{q_{k,l_k}}$. It is a Quality Sub-Characteristic weight value. It is between $(0 \leq w_{q_{k,l_k}} \leq 1)$ and represents the importance of Quality Sub-Characteristic in the set of Quality Sub-Characteristics. The sum of this value for the set of Quality Sub-Characteristics of the same Quality Characteristic has to be 1.
- $s_{q_{k,l_k}}$. It is a Quality Sub-Characteristic value. It is between $(0 \leq s_{q_{k,l_k}} \leq 1)$ because this value is actually a degree representing whether the state reached by this value satisfies or not the required model (value = 1).
- m . It is the number of Quality Sub-Characteristic values of a Feature F_i whose weight value $w_{q_{k,l_k}}$ is not 0.

A formulation is given in order to define chart representations of a set of values. The main goal of this kind of formulation is to clarify the results showed in the charts since there are several alternatives to carry out an analysis from different points of view. For instance, we can get either a general view of Features with every Feature value or a set of Sub-Features value of a specific Feature. Besides, we can have a general view about Quality Characteristics with every Quality Characteristic value or a set of Quality Sub-Characteristics value of a specific Quality Characteristic, depending on different Features or just one. Thus, this last representation case has to indicate the Sub-Features or Sub-Feature considered calculating this Quality Sub-Characteristics. The notation is explained below:

- R_F . It represents the set of Feature values $\{r_{F_1}, r_{F_2}, \dots, r_{F_n}\}$ in a general view. All Features are considered with this terminology.
- R_{F_i} . It represents a set of Sub-Feature values $\{r_{f_{i,1}}, r_{f_{i,2}}, \dots, r_{f_{i,j}}\}$ of a Feature F_i .

In the same way, in order to represent Quality Sub-Characteristic values of a Quality Characteristic, the term Q_k is equivalent to the term q_{k,l_k} .

- $S_{Q|F}$ Z. It represents a set of Quality Characteristic values $\{s_{Q_1}, s_{Q_2}, \dots, s_{Q_m}\}$ for each Quality Characteristic in a general view, according to the Sub-Feature values $\{r_{f_{i,1}}, r_{f_{i,2}}, \dots, r_{f_{i,j}}\}$ of each Feature F_i . All Features are considered.
- $S_{Q_k|F}$. It represents a set of Quality Sub-Characteristic values $\{s_{q_{k,1}}, s_{q_{k,2}}, \dots, s_{q_{k,l_k}}\}$ of a Quality Characteristic Q_k in terms of Sub-Feature values $\{r_{f_{i,1}}, r_{f_{i,2}}, \dots, r_{f_{i,j}}\}$ of each Feature F_i .
- $S_{Q_k|F_1, F_2, \dots, F_n}$. It represents a set of Quality Sub-Characteristic values $\{s_{q_{k,1}}, s_{q_{k,2}}, \dots, s_{q_{k,l_k}}\}$ of a Quality Characteristic Q_k according to a set of Sub-Feature values $\{r_{f_{i,1}}, r_{f_{i,2}}, \dots, r_{f_{i,j}}\}$ of the Features F_i considered.
- $S_{Q_k|F-F_t}$. It represents a set of Quality Sub-Characteristic values $\{s_{q_{k,1}}, s_{q_{k,2}}, \dots, s_{q_{k,l_k}}\}$ of a Quality Characteristic Q_k in terms of the Sub-Feature values $\{r_{f_{i,1}}, r_{f_{i,2}}, \dots, r_{f_{i,j}}\}$ of each Feature F_i , without considering the Sub-Feature values of the Feature F_t .

This Mol could be used to optimize the costs of implementing several new Features and Sub-Features in a methodology. Designers could predict the costs of implementing a new Feature in their methodologies. For instance, they can decide to implement the minimal number of Features and Sub-Features which have influence on a major number of Quality Characteristics and Quality Sub-Characteristics.

5. Identifying Quality Characteristics and Quality Sub-Characteristics for MDWE

In QuEF, the Technological Watch Management is a process of the Strategy phase used for the observation of trends and technologies on the rise. It also assesses the state of the art of the domain. The results of this process are used to capture information for the Portfolios. In these lines, the Portfolio of Quality Characteristics is

an artifact of the Quality Model Strategy phase that defines all important aspects that designers have to guarantee to users of these methodologies. Users, on their part, have to identify the importance of these elements in the future, bearing in mind the removal or insertion of more Quality Characteristics, depending on the current trends captured by the Technological Watch Management.

As far as Quality Characteristics are concerned, the methodology can be evaluated as a product itself. For instance, we can evaluate either Usability or Functionality of a methodology. Depending on its nature, each item is evaluated through a set of appropriate Quality Characteristics. As there is no standard glossary for MDWE terminology, it is necessary to standardize the terminology to improve the access channel for communication in MDWE.

This paper presents a set of Quality Characteristic and Quality Sub-Characteristic based on ISO/IEC 25000. Consequently, methodologies are evaluated as products. To this end, a set of Quality Characteristics, based on current literature such as ISO/IEC 9126, ISO/IEC 25000 (SQuaRE), IEEE [22] and other standards adapted to MDWE methodologies are being identified and hierarchically classified. ISO/IEC 2501n standard (Quality Model Division) that conforms this division presents a detailed quality model including characteristics for internal and external quality in use. Furthermore, the internal and external software quality characteristics are organized into sub-characteristics. Practical guidance on the use of the quality model is also provided. Quality Characteristics are classified according to ISO/IEC ISO/IEC 2501n into three different groups:

- External Quality, which measures the software (ISO/IEC 2501n).
- Internal Quality, which measures the system behavior (ISO/IEC 2501n).
- Quality in Use, which measures the effect of using the software in a specific context (ISO/IEC 2501n).

Thus, the Quality Characteristics for MDWE methodologies could be classified into two categories:

- External/Internal Quality, which measures the approach features and their behavior. They are Usability, Functionality, Reliability, Maintainability and Portability.
- Quality in Use, which measures the effect of using the approach features in a specific context. They are Effectiveness, Productivity and Satisfaction.

As Fig. 4 shows, both categories have influence on each other. Moreover, the Quality in Use category of a Web Application depends on the Web Application External and Internal Quality category.

We have selected the above Quality Characteristics because we consider that a methodology could be evaluated as a product itself. A methodology resembles a software product in that it defines a set of methods, tools or techniques that conforms Internal Quality. On the contrary, these methodologies are implanted in a work context or environment and users behave in a different manner in relation to all these Properties depending on the work context (External Quality). Finally, we can evaluate the effects of applying the methodology in a working context (Quality in Use). Therefore, in spite of using ISO/IEC 9126 or ISO/IEC 25000, Quality Characteristics work as a basis adapted to the evaluation of methodologies. It must be noted that we do not talk about software product, but MDWE methodologies, instead.

In this line, the use of a methodology ensures output consistency, although it does not guarantee Web application quality. This is due to the fact that using a methodology does not ensure Web Application quality, but a uniformity of results. This fact may even

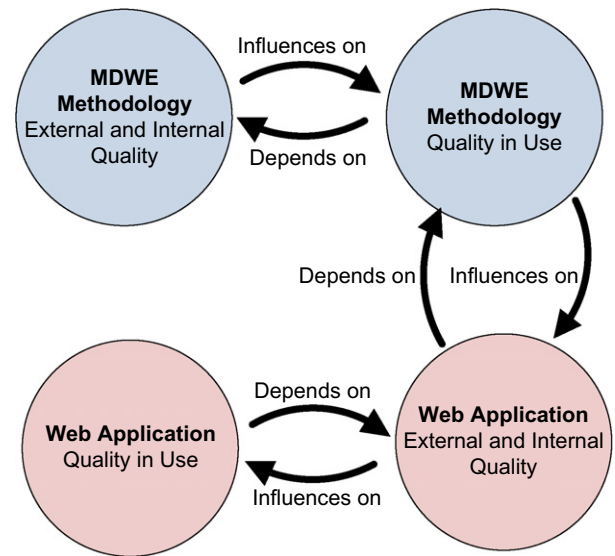


Fig. 4. Influences and dependencies between the methodology and the product.

institutionalize the creation of inappropriate Web Applications. Thus, in future work, application features will be related to methodology features in order to define how each methodology feature influences each Web Application feature. For this reason, we are working on the analysis of efficiency and productivity with the aim of relating these methodologies methods to Web Applications. However, the first step must deal with characterizing these methodologies to know what Properties must be offered to users and what Quality Characteristics must be guaranteed according to users' needs. In this sense, it is essential to describe the Properties of Web Application and the relation between Properties of methodologies and Properties of Web Applications in the results. They must also be related to Properties of methodologies, as it is necessary to know what Properties of methodologies have influence on Properties of Web Applications.

In this paper, each Quality Characteristic in ISO/IEC 25000 is described in relation to a software product, but, in this case, all Quality Characteristics and Quality Sub-Characteristic are described in relation to approach characteristics as follows:

Usability: In ISO/IEC 25000, it is a Quality Characteristic defined as: "The capability of the software to be understood, learned, used and be attractive to the user when used under specific conditions". This definition could be adapted to more closely fit our specific domain as: "The capability of an approach feature to be understood, learned, used and attractive to user when applied in specific conditions" or in a general way as: "A set of attributes that bears influence on the effort needed for use, and on the individual assessment of such an use, by a stated or implied set of users".

A set of Quality Sub-Characteristics is identified for every Quality Characteristic. Quality Sub-Characteristics associated to each Quality Characteristic of Usability are similarly described by adapting other definitions from ISO/IEC, IEEE, other standards and work already published. Quality Sub-Characteristics of Usability are described as:

- **Learnability.** The capability of an approach feature to enable the user to learn how to use it. [Adapted from ISO/IEC 25000].
- **Understandability/ Comprehensibility.** The capability of being understood and the extent to which an approach feature is clear, without ambiguity, and easily comprehensible. [Adapted from ISO/IEC 25000].
- **Simplicity.** The degree to which an approach feature has a straightforward design easily understood. [Adapted from IEEE].

- **Interpretability.** The extent to which an approach feature is organized into suitable units of information for the user's skill.
- **Operability/ Ease of Operation.** The capability of an approach feature to enable the user to operate and control it. [Adapted from ISO/IEC 25000].
- **Attractiveness.** The extent to which an approach feature is sufficiently attractive for users.

Functionality: In ISO/IEC 25000, it is a Quality Characteristic defined as: “*The capability of a software product to provide functions which meet stated and implied needs when the software is used under specific conditions*”. This definition could be adapted to more closely fit our specific domain: “*The capability of an approach feature to provide functions which meet stated and implied needs when the methodology is used under specific conditions*” or in a general way as: “*A set of attributes that bears on the existence of a set of functions and their specific Properties. The functions are those that satisfy stated or implied needs*”. Quality Sub-Characteristics of Functionality are described as follows:

- **Suitability.** The capability of an approach feature to provide an appropriate set of functions for specific tasks and users' objectives. [Adapted from ISO/IEC 25000].
- **Accuracy.** The capability of an approach feature to provide the right or agreed results or effects with the needed degree of precision. [Adapted from ISO/IEC 25000].
- **Interoperability.** The capability of an approach feature to interact with one or more specific approach features. [Adapted from ISO/IEC 25000].
- **Compliance.** The capability of an approach feature to adhere to standards, conventions or legal regulations and similar prescriptions. [Adapted from ISO/IEC 25000].
- **Interactivity.** The extent to which an approach feature is used or can be adapted to one's personal preferences through interactive elements.
- **Applicability.** The extent to which an approach feature is unique, useful and easily applicable for the target community.
- **Accessibility:** The extent to which an approach feature provides support and mechanisms and makes users obtain their desired documentation or tools faster and more easily.
- **Flexibility.** The extent to which an approach feature is expandable, adaptable and easily applied to other needs.
- **Traceability.** The capability of an approach feature to provide an appropriate set of functions that enables models and code to be well-documented, verifiable and easily attributed to a source.
- **Transformability.** The capability of an approach feature to provide an appropriate set of functions to transform models into other models or codes.
- **Testability.** The capability of an approach feature to provide an appropriate set of functions for testing. [ISO/IEC 25000].

Portability: In ISO/IEC 25000, it is a Quality Characteristic defined as: “*A set of attributes that bears on the ability of software to be transferred from one environment to another*”. This definition could be adapted to more closely fit our specific domain: “*The ease to transfer an approach feature from one environment to another*” or in a general way as: “*A set of attributes that bears on the ability of an approach feature to be transferred from one environment to another*”. Quality Sub-Characteristics of Portability are described as follows:

- **Installability.** The capability of an approach feature to be installed in a specific environment. [ISO/IEC 25000].
- **Replaceability [ISO/IEC 25000]/Reusability [IEEE 610].** The capability of an approach feature to be used in place of another specific approach for the same purpose in the same environment.

- **Adaptability.** The capability of an approach feature to be adapted to different specific environments without applying actions or means other than those provided for this purpose to the considered approach. [ISO/IEC 25000].

Reliability: In ISO/IEC 25000, it is a Quality Characteristic defined as: “*The ability of the software product to perform its required functions in stated conditions for either a specific period of time or number of operations*”. This definition could be adapted to more closely fit our specific domain: “*The ability of a feature approach to perform its required functions in stated conditions for either a specific period of time or number of operations*” or in a general way as: “*A set of attributes that bears on the capability of an approach feature to maintain the level of performance under set conditions for a determined period of time*”. Quality Sub-Characteristics of Reliability are described as follows:

- **Maturity.** The capability of a feature approach to avoid failure as a result of different defects in the approach.
- **Recoverability case of failure.** The capability of an approach feature to re-establish a specific level of performance and recover a stated development affected point.
- **Fault Tolerance:** The capability of an approach feature to maintain a specific level of performance in cases of faults (defects) or requirements infringement.
- **Availability:** The degree to which an approach feature is operational and accessible when used.
- **Currently:** The extent to which an approach feature provides updated information.
- **Compactness:** The state of being compact. The closer to a specific domain, the more compact an approach feature becomes.
- **Relevancy:** The extent to which an approach feature is applicable and helpful for users' needs.

Maintainability: In ISO/IEC 25000, it is a Quality Characteristic defined as: “*A set of attributes that bears on the effort needed to make specific modifications*”. This definition could be adapted to more closely fit our specific domain: “*The ease to modify an approach feature to: correct defects, meet new requirements, facilitate future maintenance or cope with a changed environment. These activities are known as methodology maintenance*” or in a general way as: “*A set of attributes that bears on the effort needed to make specific modifications. The ease to modify an approach feature to correct defects, meet new requirements, make future maintenance easier or adapted to a changed environment.*” Quality Sub-Characteristics of Maintainability are described as follows:

- **Stability:** The capability of an approach feature to avoid unexpected effects from approach modifications. [ISO/IEC 25000].
- **Analyzability:** The capability of an approach feature to found deficiencies or causes of failures in the approach, or in the identified parts to be modified. [ISO/IEC 25000].
- **Changeability:** The capability of an approach feature to enable specific modifications. [ISO/IEC 25000].
- **Testability:** The capability of an approach feature to enable a modified approach to be tested. [ISO/IEC 25000].

6. Sample application

6.1. The NDT methodology

NDT (Navigational Development Techniques) [15,16,36] is a Model-Driven Web methodology initially defined to deal with requirements on web development. NDT starts with a goal-oriented phase of requirements and defines a set of transformations to

generate analysis models. NDT has evolved in the last years and offers a complete support for the whole lifecycle. There are different papers published about NDT methodology. IEEE TSE [15], for example, presents it in details. NDT methodology covers viability study, requirements treatment, analysis, design, construction, implementation, as well as maintenance and test phases as software development phases. Additionally, it supports a set of processes to bear out project management and quality assurance. In the last year, it evolved to support different lifecycles: sequential, iterative and agile processes.

Nowadays, NDT is supported by NDT-Suite, which offers a tool support for each phase of the complete lifecycle of a software project. NDT-Suite consists in a set of tools to apply the NDT Methodology in practical environments. The suite of NDT not only focuses on providing support to project development by means of the methodology, but also gives support to verify and validate the work with NDT. The toolkit of NDT-Suite is composed of the following tools:

- **NDT-Profile.** It is a profile defined in Enterprise Architect. It offers a set of tools that allows us to define methodology artifacts.
- **NDT-Driver.** It automatically executes the transformations defined in NDT methodology, taking a project developed by NDT-Profile as input. For practical purposes, this tool reduces the time spent on the models description and design phase after the requirements phase, since the basic models obtained by this tool provides the analyst with a starting point.
- **NDT-Quality.** It is a tool that, starting with a NDT-Profile file, checks if NDT rules and relations are correctly followed.
- **NDT-Prototypes.** It is a tool that automatically generates a set of XHTML prototypes from the navigation models described in the analysis phase of a project developed with NDT-Profile. For practical purposes, this tool reduces the time spent on producing a Web Application as the development team provides a starting point when conducting the system construction.
- **NDT-Glossary.** It is a tool that implements an automatic procedure and generates the first instance of the glossary of terms for a project by means of NDT-Profile tool.

In the last 10 years, NDT and NDT-Suite have been applied in a high number of real projects in Spain. In fact, they are currently used in several projects carried out by different companies, either public or private, big or small. Today, a high number of Web systems with different providers, users or development teams are working with them. Moreover, since 2004, an important project is being developed in liaison with the Andalusian Regional Cultural Ministry [9]. Another important project is being carried out in liaison with Emasesa [13], where the AQUA-WS project (AQUA-Web-Services) was finished in 2011. NDT was also widely applied in the e-health environment. In 2006, Alcer Foundation [3] used it within the system to manage patients' degree of handicap.

6.2. Checklists and properties applied to NDT methodology

The following example presents the application to evaluate NDT and NDT-Suite. Functionality, Reliability, Portability, Usability and Maintainability Quality Characteristics have been studied in an implementation process using Microsoft Excel. This example shows Quality Sub-Characteristics of Functionality together with the relationship with different Properties such as MDE, Experience, Tool Support and Web Modeling. Studying Properties constitutes the objective of another work, although Properties are briefly summarized below:

- **MDE:** This Feature describes the specific MDWE aspects. It covers some aspects defined in the approach, such as the evaluation of Web Application development process, specific

conceptual levels for this domain and levels of abstraction. MDE Feature is more general than Web Modeling Feature and has been structured with these Sub-Features: Levels of abstraction, Standard definition, Model-based testing, Transformations and Traces. Each of them describes some aspects related to their names.

- **Experience:** This Feature describes the maturity state of a methodology. For example, it deals with the year the approach was introduced, the number of modeling examples or the number of applications in real-world projects. Thus, this Feature has been structured into several Sub-Features: Topicality, Modeling examples, Application in real-world projects, Publications and External web references. Each of them describes aspects related to their names.
- **Tool Support:** This Feature is used for specifying whether or not it provides a Tool such as the creation tool, edition tool or other different tools. This Feature has been divided into several Sub-Features: Creation, Edition and Composition tool support, Analysis tool support, Transformation tool support, Code generation and specific platform tool support, Trace tool support and Team work tool support. Each of them describes some aspects related to their names.
- **Web Modeling:** This Feature describes MDE aspects dealing with the Modeling language definition used, such as Suitability for the MDWE domain, Complexity, Transformations, Traces, Test cases and Rule Generation models, as a prior requirement to successfully employ MDE in the MDA style of OMG. It has been structured into several Sub-Features: Web conceptual levels, Interfaces, Development process, Content Feature modeling, Presentation Feature modeling, Navigation Feature modeling and Business Feature modeling. Each of them describes aspects related to their names.

Table 2 shows an example of MDE Sub-Feature Checklist applied to NDT methodology. Actually, this is a Checklist generated in terms of the Quality Model definition in the Strategy and Design Phase. The Checklist is used in the Operation Phase. The objective of using these Checklists is to capture all Properties that have been implemented in the methodology managed at that moment. All these Properties have previously been selected by conducting a Strategy phase based on a Systematic Literature Review process that involves a large initial effort and includes the study of a variety of MDWE methodologies. In this sense, a consensus reaching process must be carried out to get a common description of these Properties for the domain description of MDWE methodologies.

In these lines, Properties defined are very complete and we have decided to leave users the possibility of deciding on the importance of each Property, Sub-Features and Features, given both the immaturity of the domain and the lack of the consensus decision reached by users.

In these lines, in Domínguez-Mayo et al. [12] a method is defined based on the essence of the analytic hierarchy process. In this sense, a methodology designers group has to reach a consensus on the properties of MDWE methodologies. Thus, in the Operation phase, users applying weight value can define the importance of each Property.

It is worth recalling, as stated in the previous section, that it is hard to reach a consensus domain with agreed elements to get a common Quality Model. Besides, it is even more difficult in the Strategy phase because it not only has to involve all users, but also the domain remains immature. Once the domain becomes more mature all consensus aspects pass onto the Strategy phase for consideration. Consequently, if users consider that all Properties have the same degree of importance in the Operation phase, then they are evaluating the totality of their methodologies. Nevertheless, if they think that some Properties are more important than others,

Table 2
standard definition Sub-Feature of the MDE Feature Checklist.

PT-<221> metamodel, schema, grammar or ontology	PB-<2211>	It provides a metamodel based on Meta Object Facility (MOF) and XML-based exchange format for UML and other MOF-based metamodels and models (XMI)	Supported
	PB-<2212>	It uses a metamodel based on the MOF 1.4 or 2.0 (current version) and XMI 2.0 or 2.1 (current version)	Supported
	PB-<2213>	It provides a UML-profile (a metamodel extended from the standard UML metamodel)	Supported
	PB-<2214>	It uses UML 2.0 (current version) for defining metamodels	Supported
	PB-<2215>	It uses a standard language for metamodels consistency, such as Object Constraint Language (OCL) in order to provide constraints and object query expressions on any MOF metamodel that cannot otherwise be expressed by diagram notation	Supported
	PB-<2216>	It provides BPMN metamodel for modeling business process	Not supported
	PB-<2217>	It uses BPMN 1.2 or 2.0 (current version) for defining metamodels	Not supported
PT-<222> model or visual syntax	PB-<2221>	It provides UML or a similar standard model for Model-Driven Engineering	Supported
	PB-<2222>	It uses UML 2.0 (current version) for defining models	Supported
	PB-<2223>	It uses a standard language for models consistency, such as Object Constraint Language (OCL) in order to provide constraints and object query expressions on any MOF model that cannot otherwise be expressed by diagram notation	Supported
	PB-<2224>	It provides BPMN or a similar standard model for modeling business process	Not supported
	PB-<2225>	It uses BPMN 1.2 or 2.0 (current version) for defining models	Not supported

they can change them in the Operation phase by means of weight values.

The standard definition of Sub-Feature describes whether the Property is elaborated or not with standard notations. Properties can be identified as Derived Properties (which are known as PD-<Identification number>) or as Base Properties (which are known as PB-<Identification number>). A Derived Property has a set of Base Properties. Besides, they are hierarchical as PT-<Identification number>. This Sub-Feature evaluates whether a Web modeling language has been explicitly defined in terms of a metamodel (including UML profiles), a grammar and a semantic description according to semantic Web technologies, or if such a definition is absent.

In the Checklist, each Property value can represent either a quantitative or qualitative value. This example only shows qualitative values and they can be *Supported*, *Partly Supported* or *Not Supported* depending on the degree in which the methodology implements the presented Property. In the next step, each qualitative Property is changed into a quantitative value that should be normalized.

Thus, Table 3 shows the quantitative values for the example offered in Table 2. Table 3 shows the Checklist values for the standard definition of Sub-Features, although they have been transformed into quantitative values. This formulation is being currently defined in other work.

The complete Checklist has been applied to NDT methodology. Table 4 shows the total values for each MDE Sub-Feature. These values are applied in formula (1) to calculate Quality Sub-Characteristic values, as presented in Section 5.

Table 3
Values of the Checklist of the Standard definition Sub-Feature.

PT-<221> metamodel, schema, grammar or ontology	PB-<2211>	1
	PB-<2212>	1
	PB-<2213>	1
	PB-<2214>	1
	PB-<2215>	1
	PB-<2216>	0
	PB-<2217>	0
PT-<222>model or visual syntax	PB-<2221>	1
	PB-<2222>	1
	PB-<2223>	1
	PB-<2224>	0
	PB-<2225>	0
		2/3

Table 4
Total values of the Sub-Features of MDE.

Sub-Feature name	Values
Levels of abstraction	1
Standard definition	2/3
Model-based testing	1
Transformations	3/4
Traces	1
	7/8

Table 5
Features total values.

Feature name	Values
MDE	7/8
Experience	1/4
Tool support	7/8
Web modeling	3/4

Consequently, NDT methodology has been applied for each completed Feature Checklist. Table 5 shows the Features total values obtained for each Feature. These values are also useful, since they give an idea of the degree in which the reached state of a methodology implements the required model. This fact is being currently studied in other work.

6.3. Filling association link values in the MoIs (Matrix of Influences)

In this phase, the association links between Sub-Features and Quality Sub-Characteristics are defined in a Matrix of Influences (MoIs). On the one hand, a set of Properties (Hierarchical by Features and Sub-Features) are determined and, on the other hand, a set of Quality Characteristics (Hierarchical by Quality Characteristics and Sub-Characteristics) are defined. Thus, a set of hypotheses is proposed to indicate which Quality Sub-Characteristic is affected by each Sub-Feature. For example, Functionality is described as a set of Quality Sub-Characteristics that could be affected by one of the different Sub-Features, as shown in Table 6. This table only represents whether a Sub-Feature influences or not a Quality Sub-Characteristic.

Table 6
Matrix of Influences between Sub-Features and Quality Sub-Characteristics of Functionality.

		Functionality										
		Suitability	Accuracy	Interoperability	Compliance	Interactivity	Applicability	Accessibility	Flexibility	Traceability	Transformability	Testability
MDE	Levels of abstraction	✓	✓	✓	✓	✓	✓	×	✓	×	×	×
	Standard definition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Model-based testing	✓	✓	×	×	×	×	×	×	×	×	✓
	Transformations	✓	✓	✓	×	✓	✓	×	×	×	✓	×
	Traces	✓	✓	×	×	×	×	×	×	✓	×	×
Experience	Topicality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Modeling examples	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Application in real-world projects	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Publications	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tool Support	External Web references	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Creation, edition and composition tool support	✓	✓	×	✓	✓	✓	×	✓	×	×	×
	Analysis tool support	✓	✓	×	✓	✓	✓	×	✓	×	×	✓
	Transformation tool support	✓	✓	✓	✓	✓	✓	×	✓	×	✓	×
	Code generation and specific platform tool support	✓	✓	✓	✓	✓	✓	×	✓	×	×	×
	Trace tool support	✓	✓	×	×	✓	✓	×	✓	✓	×	×
	Team work tool support	✓	✓	×	×	✓	✓	✓	✓	×	×	×
Web Modeling	Web conceptual levels	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×
	Interfaces	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
	Development process	✓	✓	✓	✓	✓	✓	×	✓	×	×	×
	Content Feature modeling	✓	✓	✓	✓	✓	✓	×	×	×	×	×
	Presentation Feature modeling	✓	✓	✓	✓	✓	✓	×	×	×	×	×
	Navigation Feature modeling	✓	✓	✓	✓	✓	✓	×	×	×	×	×
	Business Feature modeling	✓	✓	✓	✓	✓	✓	×	×	×	×	×

As it is not still clear up to what degree Properties (categorized by Features and Sub-Features) exert an influence on Quality Characteristics, we have decided to consider, as a first step in our application example, whether there is an existing influence between both in the Mol. If a Sub-Feature has an influence on a Quality Sub-Characteristic, then, the degree value is 1 and if a Sub-Feature has no influence on a Quality Sub-Characteristic, then the degree value is 0. This scale of influence has to improve by elaborating experiments to define and enhance the scale of influence for these association links.

The definition of these association link values are not defined as an active Strategy phase, but as an active Operation phase in QuEF, since it is difficult to reach a consensus on the influence of Properties on Quality Characteristics. Then, each Quality Characteristic is defined in terms of all Properties that influence it. The usefulness or profitability of this matrix is due to the fact that it is within this matrix that Quality Characteristics brings out their meaning. "Meaning" is understood as the possibility of relating Quality Characteristics to Properties because this relation provides all the necessary information to calculate a value.

Thus, there are only two degrees of influence defined in this case; a Sub-Feature influencing a Quality Sub-Characteristic (\checkmark) or not (\times). These values are transformed to quantitative values between 0 and 1 in order to calculate the formula 1. The application of these different Features and Sub-Features could provide the approaches with a wider application base. Every Quality Sub-Characteristic of Functionality and their relationships are described below as an example. Our initial hypothesis is that Properties could have influence on:

- **Suitability.** The more Sub-Characteristics an approach provides, the more suitable it is, since each Sub-Characteristic provides a set of functions for specific tasks and users' objectives.
- **Accuracy.** Every Sub-Characteristic specifies the use of an approach, because they (different tools with different objectives) can generate, analyze and edit models and codes in an automatic way.
- **Interoperability.** If an approach provided tools to manage rules for the Transformations and Composition Sub-Characteristic(s), it would be more interoperable as different models from other approaches would be generated from those tools.
- **Compliance.** Every Sub-Characteristic is conceived as a tool helping models and codes development in an automatic way, following conventions and similar prescriptions and assuming standards.
- **Interactivity.** Every Sub-Characteristic is a tool that can be adapted to the user's personal preferences through interactive elements.
- **Applicability.** The use of every tool is specific, useful and easily applicable by the target community. (para o por la comunidad.
- **Accessibility.** The Repository tool is the only one that provides users with support and mechanisms to reach the desired documentation or tools in a faster and easier way.
- **Flexibility.** Tools can help approaches be more expandable, adaptable and easily applied.
- **Traceability:** Trace tools give models and codes appropriate sets of functions to be well-documented, verifiable and easily attributed to a source.
- **Transformability.** Both, Transformation tool Sub-Characteristic and Rule tool Sub-Characteristic add an appropriate set of functions for transforming models to other models or codes.
- **Testability.** The Analysis tool and Test tool offer a relevant set of functions to check metamodels, models and codes.

6.4. NDT methodology quality evaluation

The NDT methodology has been analyzed from the point of view of Quality Characteristics and Quality Sub-Characteristics, consider-

ing several and different set of Properties. Nevertheless, to restrict the content of this paper, we will only analyze Functionality values.

The validation of results is determined by the experimentation and experience of a defined relation between Properties and Quality Characteristics in the Matrix of Influences (Mols). In that Matrix, Properties are associated with Quality Characteristics to indicate their influence on Quality Characteristics. For instance, UML is a standard notation defined by OMG that according to experience influences Usability of methodologies. Thus, if a methodology implement UML diagrams, then it has a Property that influences Usability, being an approach easy to understand, learn, use and attractive to users. Features and Quality Characteristic relations have to be improved continuously with empirical experiments.

Fig. 5 consists of a gray line representing the required model of Functionality Quality Characteristic. The black-pointed line represents Functionality in NDT methodology.

Fig. 5 reveals the following values $A_{Q^{Functionality}/F_{MDE}}$, which means that a set of Quality Sub-Characteristic Functionality values is shown in terms of MDE. Table 7 includes Quality Sub-Characteristic values calculated by formula (1) explained in Section 5. It can be noticed that if NDT methodology only considers the MDE Feature, it betters the Suitability and Accuracy scores. This means that this Feature improves the methodology, providing a set of functions for specific tasks and users' objectives and specifies the approach. Accesibility is a Quality Sub-Characteristic that NDT methodology should improve for MDE Feature.

Therefore, the methodology does not give support nor mechanisms related to MDE, so users cannot reach their desired documentation faster and more easily. It is recommended that designers improve this Quality Sub-Characteristic for the MDE Feature. Fig. 6 shows Functionality in terms of all Features, as defined

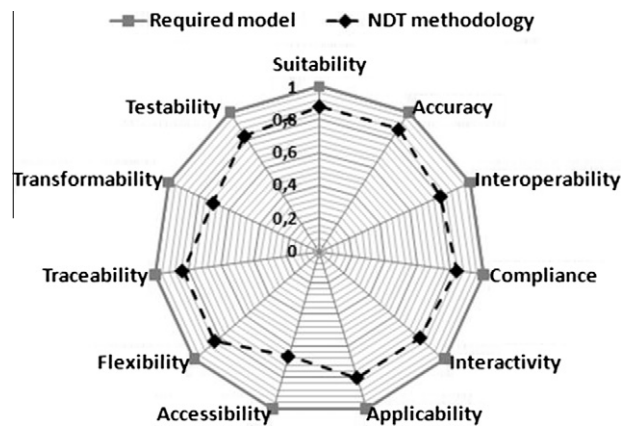


Fig. 5. Quality sub-characteristics of functionality, but only in terms of MDE.

Table 7
Quality Sub-Characteristic Functionality values, but only in terms of MDE.

Quality Sub-Characteristic name	Values
Suitability	7/8
Accuracy	7/8
Interoperability	4/5
Compliance	5/6
Interactivity	4/5
Applicability	4/5
Accessibility	2/3
Flexibility	5/6
Traceability	5/6
Transformability	5/7
Testability	5/6

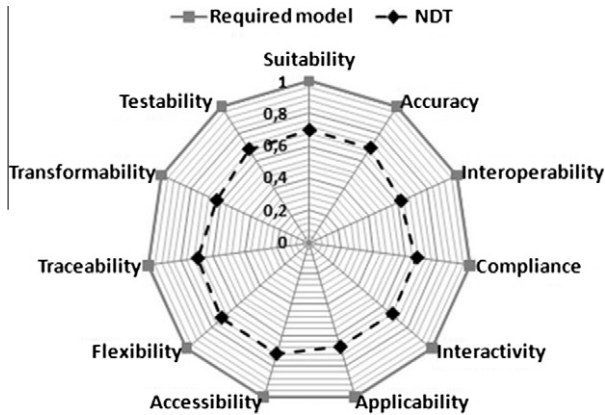


Fig. 6. Quality sub-characteristics of functionality in terms of all features.

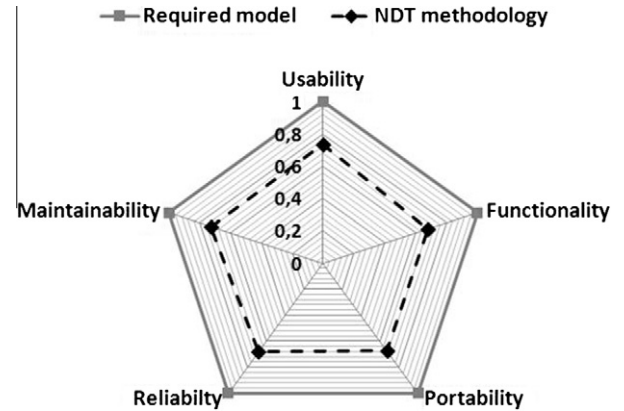


Fig. 7. Quality characteristics in terms of all features.

in the following formula $S_{Q_{Functionality}|F}$. In general, NDT methodology has uniform scores for every Quality Sub-Characteristic of Functionality. Table 8 shows these values.

As it can be observed, results are uniform for this set of Quality Sub-Characteristics, mainly due to their similarity or the need of more Features and Sub-Features to identify differences among these Quality Sub-Characteristics, whereby Feature Checklists would have to be descriptively defined in detail. Furthermore, it can get better scores for every Quality Sub-Characteristic of Functionality because the reached state is far from the required Quality Model. However, it can also be observed that NDT methodology has best scores in Accesibility and Flexibility when all Features are considered. Therefore, it implies that in general (not in cases where only MDE is considered), NDT provides support and mechanisms to make users reach their desired documentation faster and more easily. Additionally, the methodology becomes more expandable, adaptable and easy to use. Nevertheless, the NDT methodology has to improve Quality Characteristics in general, since the reached state is far from the required model.

Fig. 7 deals with a general view of Quality Characteristics in terms of all Features, as represented in the following formula $S_{Q|F}$. Table 9 shows the values that have been calculated using formula (2), explained in Section 5. In this figure, it is realized that, in general, NDT methodology can be improved for every Quality Characteristic because it is far from the required Quality Model.

Besides, NDT methodology has best scores in Usability and Maintainability, what reveals that this methodology has the capability of being understood, learned, used and it is attractive to users when applied under specific conditions. It can be modified, if needed, to correct defects, meet new requirements, make future maintenance easier or cope with a changed environment. In other words, NDT methodology implements Features and Sub-Features

Table 9

Quality Characteristic values in terms of all Features.

Quality Characteristic name	Values
Usability	3/4
Functionality	2/3
Portability	2/3
Reliability	2/3
Maintainability	5/7

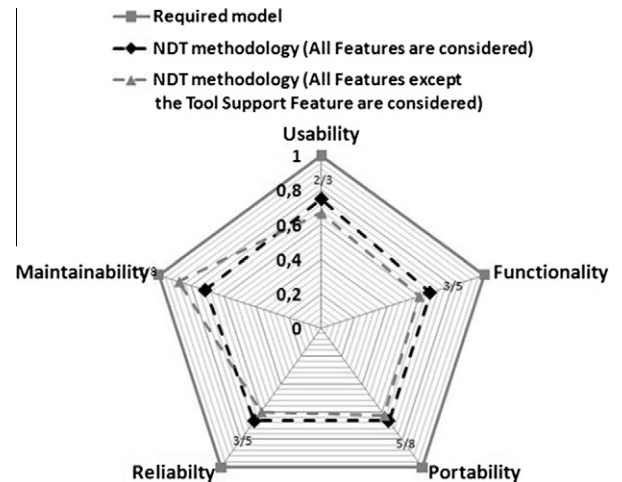


Fig. 8. Quality characteristics in terms of all features except tool support.

Table 8

Quality Sub-Characteristic Functionality values in terms of all Features.

Quality Sub-Characteristic name	Values
Suitability	2/3
Accuracy	2/3
Interoperability	5/8
Compliance	2/3
Interactivity	2/3
Applicability	2/3
Accessability	5/7
Flexibility	5/7
Traceability	2/3
Transformability	5/8
Testability	2/3

that influence these Quality Characteristics. However, it must improve other Quality Characteristics, such as Functionality, Portability and Reliability.

Finally, Fig. 8 shows a comparison between the results in Fig. 7, a general view of Quality Characteristics in terms of all Features (in black-pointed line) and a general view of Quality Characteristics in terms of all Features, without considering Tool Support (in gray-pointed line), as represented in the following formulas $S_{Q|F_{MDE \cup F_{WebModeling} \cup F_{Maturity}}}$ or $S_{Q|F - F_{ToolSupport}}$.

Table 10 shows the values that have been calculated by means of formula (2). This table points out Quality Characteristics in terms of all Features (values 1) and all Features, without considering the Tool Support (values 2). The results are lower in some Quality Characteristics because the Tool Support Feature has influenced some Quality Characteristics such as Usability, Functionality, Portability and Reliability.

Table 10

Quality Characteristic values in terms of all Features (values 1) and all Features except Tool Support (values 2).

Quality Characteristic name	Values 1	Values 2
Usability	3/4	2/3↓
Functionality	2/3	3/5↓
Portability	2/3	5/8↓
Reliability	2/3	3/5↓
Maintainability	5/7	7/8↑

This development occurs because NDT methodology implements those Sub-Features of Tool Support that affect these Quality Sub-Characteristics. Nevertheless, values 2 are closer to values 1 and the reached state is not the required model. Thus, the designers could highly improve NDT methodology. If the Tool Support Characteristic were not considered (Fig. 7), it would be observed that this Feature would get lower values for Maintainability. The cause could be that not all features defined by NDT methodology are implemented by the Tool support. In other words, the Tool Support Feature includes aspects that NDT methodology does not implement. For this reason, the results are lower without considering this Feature. In this sense, the framework could be used for predicting and simulating different situations and designers could decide what features should be implemented, attending to the analyzed results. This is an important aspect, as they can optimize the implementation cost of those features that highly affect the Quality Characteristic that must be firstly implemented.

Generally, NDT methodology has sensitively best scores in Usability, being an approach easy to understand, learn, use and attractive to users. Additionally, it also has best scores in Maintainability, being an easy approach to be modified when correcting defects, meeting new requirements, making future maintenance easier or coping with a changeable environment.

Besides, it shows good performance in Functionality, Reliability and Portability, although it is less sensitive to Quality Characteristics such as Usability and Maintainability. Finally, it could be improved in all Quality Characteristics due to the required Quality Model.

7. Future work

Although Microsoft Excel was used to implement the sample application, we are currently working on QuEF-TS, a general tool support for implementing the QuEF framework. This tool support is composed of different tools that are focused on each phase of QuEF (Strategy phase, Design phase, Transition phase, Operation phase and Quality Continuous Improvement phase):

- QuEF-S: It is the tool that supports the Quality Model Strategy phase. (It is being developed).
- QuEF-D: It is the tool that supports the Quality Model Design phase. (It is not developed yet).
- QuEF-T: It is the tool that supports the Quality Model Transition phase. (It is not developed yet).
- QuEF-O: It is the tool that supports the Quality Model Operation phase. (It is being developed).
- QuEF-QCI: It is the tool that supports the Quality Continual Improvement in quality management. (It is not developed yet).

QuEF-TS can generate all artifacts (for instance the Checklists and MoIs) in terms of the Quality Model specified in the Strategy phase and Design phase with QuEF-S and QuEF-D tools. Thus, an important issue is to define Quality Model to be used in the Operation phase with QuEF-O, which contains specific Checklists and all the necessary elements based on the Quality Model to be analyzed, controlled and evaluated for this type of methodologies. If some changes have to be applied in the Quality Model, the QuEF-T is

the tool to be used for applying the Transition phase without affecting the quality management in the Operation phase. Finally, QuEF-QCI is a tool that supports the Quality Continuous Improvement phase of QuEF. Using this tool support could help improve efficiency not only in methodologies, but also in other contexts, since this approach enhances the understanding of strengths and weaknesses in different contexts. In this sense, we are working to apply QuEF-TS in other contexts. The specific contexts have only to be defined in the Quality Model. QuEF-TS generates Checklists and MoIs to start with the Operation phase (analysis, control, and evaluation).

As a future work, we are going to use QuEF-TS to work with other contexts like CMMI [8] and standards like ISO/IEC 15504 and ISO/IEC 12207. This use of the tool would help companies implement and manage their capability and maturity. Another specific context to apply QuEF is the Green IT aspects for sustainability management and costs reduction in companies as well as the implementation of different agile software development methodologies like XP (Extreme Programming), Scrum or RUP (Rational Unified Process).

Nowadays, we are also working in the Strategy and Design phase with specific strategic models to establish some criteria for a methodology designers group and reach a consensus on the definition of several aspects, such as the Properties hierarchy or Quality Characteristics, despite the importance of these elements and the influence of Properties on Quality Characteristics. Thus, these methods could help define the weights for Properties, Quality Characteristics and Quality Sub-Characteristics. Furthermore, this process would help reach a consensus on the definition of Sub-Features influence on Quality Sub-Characteristics.

In this line, a first prototype of QuEF-S (tool support for the Strategy phase) for a first statistical study is being currently developed to gather the general opinion of MDWE experts, designers and users. We are using a fuzzy group analytical hierarchy process (AHP) [44] approach to manage this task. In these studies, a set of web pages for a first statistical study is being currently developed to know the opinion of MDWE experts, designers and users. In this statistical study, designers and users have to set the necessary weight to consider a Quality Characteristic, Quality Sub-Characteristic, Feature and relevant Sub-Feature. The weight would indicate the degree of importance.

Another objective is that QuEF-TS is connected to well known social networks like Facebook [17] or Twitter [46] to make the Strategy and Design phase easier and reach the real quality management objectives. The use of social networks does not only involve the number of people that already use them, but it entails the fact that these methodologies are carried out by people and these type of systems make easy the communication among them. Besides, lots of methods in the Strategy phase of QuEF include statistical methods focused on stakeholders in order to detect the real needs and demand as well as the quality aspects that have to be guaranteed. Social networks, in particular Facebook and Twitter, offer a complete API that permit developers to improve the connectivity of stakeholders in order to better the communication and setting statistical methods for the Strategy phase and the final design of the Quality Model in the Design phase of QuEF. We think that having access to social networks results crucial to execute an effective quality management in the future.

8. Conclusions

A set of Quality Characteristics and Quality Sub-Characteristics have been proposed according to ISO/IEC 9126, SQuaRE (ISO/IEC 25000), IEEE and other standards which have been adapted to MDWE standard in order to manage quality in MDWE methodolo-

gies. Although there are some work to converge towards a common metamodel to achieve the interoperability of these methodologies, this kind of methodologies has different metamodels and models, transformations and tool support that work in a different way. Even though users are the design target for these approaches, this situation does not let them know how to use these approaches since they do not know how these approaches can help them and which of the approaches is more suitable for them.

In addition, the diversity in some features of the design of these approaches reflects that designers of these approaches do not have a clear idea of users' needs and the Quality Characteristics they have to guarantee these needs. Therefore, designers have a twofold task; to identify the real Properties that cover users' needs and the Quality Characteristics that users demand. The convergence towards a standardization of these methods together with a consensus decision on quality management and a common Quality Model would result in that both, users and designers, apply and design these methodologies efficiently. This would be possible because they would have a common strategy, the same objectives and purposes as well as the same design of the model to be achieved. At the same time and due to their shared knowledge, designers would have a clearer idea of the changes that should be performed in the Quality Model, in case of changes in technologies or trends while designing these methodologies. Users will have the chance of comparing methodologies, and designers, on their part, will have the chance of controlling and improving the design of the methodologies.

In this line, it is necessary to standardize the terminology in order to improve the access channel for communication in MDWE. Furthermore, a brief example shows how these Quality Characteristics (Hierarchical by Quality Characteristics and Sub-Characteristics) are related to Properties (Hierarchical by Features and Sub-Features). A set of formulas has been suggested in order to analyze quality in MDWE methodologies. Additionally, we have proposed some formulas to evaluate quality in terms of different Properties and define a Matrix of Influences (Mols). As this matrix determines the influence of each Sub-Feature on each Quality Sub-Characteristics, designers are able to guess the costs of implementing a new Feature on their methodology with it. In future work, the Quality in Use is a quality aspect that has to be adapted to evaluate MDWE methodologies in the use context.

Regarding the contributions obtained from this research work, it can be stated that an environment is required either to improve the current MDWE approaches or develop new MDWE methodologies. Despite the fact that the domain is immature, we consider that using this framework to manage quality would enhance the design and methods of these MDWE approaches as a consequence of a clear strategy focused on quality continuous improvement of the Quality Model.

Another important aspect is that the use of a methodology ensures output consistency. However, it does not guarantee Web Application quality, but uniformity of results. Thus, in future work, application features will be related to methodology features in order to define how each methodology feature influences each Web Application Feature. For this reason, we are working on analyzing efficiency and productivity with the aim of relating these methodologies approaches to Web applications. Nevertheless, the first step must consist in characterizing these methodologies to identify Properties that must be offered to users and Quality Characteristics that must be guaranteed according to users' needs.

To conclude, we consider essential to satisfy user's demands and expectations. This aspect can be observed in the Quality Model and designers perfectly know the aspects that should be considered. Therefore, if users fulfil their requirements, the results obtained (Web applications) will probably improve, but always depending on the use they make of Properties. This could be interesting because if these relations are defined, it will be possible to

guess the Properties of a methodology that need to obtain a set of requirements in a Web Application. Even designers will optimize the implementations of Features and Sub-Features in their methodologies according to the results they can get. NDT methodology has been analyzed and evaluated for the Functionality Quality Characteristic in a brief example about the use of Quality Characteristics and Sub-Characteristics. We have checked that this entire environment could be also useful for either simulating different situations or making decisions on Properties designers must implement in methodologies in terms of the analyzed results.

Acknowledgements

This research has been supported by the Project QSimTest (TIN2007-67843-C06_03) and by the Tempros Project of the Ministerio de Educación y Ciencia, (TIN2010-20057-C03-02), Spain.

References

- [1] A. Abran, R.E. Al-Qutaish, J. Desharnais, Harmonization issues in the updating of the ISO standards on software product quality, *Metrics News Journal* 10(2) (2005) 35–44. (Otto-von-Guericke University of Magdeburg, Germany, December, (ISSN: 1431–8008)).
- [2] M. Azuma, SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality, *ESCOM (European Software Control and Metrics conference)*, 2001, pp. 337–346.
- [3] Alcer. Federación Nacional de Asociaciones para la lucha contra las enfermedades renales. <www.alcer.org> (retrieved 04.11).
- [4] V. Basili, H.D. Rombach, The TAME project: towards improvement-oriented software environments, *IEEE Transactions on Software Engineering* 14 (6) (1989) 758–773.
- [5] P. Becker, L. Olsina, Towards support processes for web projects, in: *Proceedings of the 10th International Conference on Current Trends in Web Engineering*, Springer-Verlag, 2010, pp. 102–113.
- [6] C. Calero, J. Ruiz, M. Piattini, Classifying Web Metrics Using the Web Quality Model. 29 (3) (2005) 227–248.
- [7] C.P. Team, CMMI for Development, Version 1.2, Preface, *Software Engineering Institute*, Carnegie Mellon University, August, 2006.
- [8] Consejería de Cultura. <<http://www.juntadeandalucia.es/organismos/cultura.html>> (retrieved 04.11).
- [9] F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías, I. Ramos, L.F. Sanz, A Quality Evaluation Framework for MDWE Methodologies, in: *Proceedings of the 18th International Conference on Software Quality Management (SQM)*, The British Computer Society, London, UK, 2010, pp. 171–184, (ISBN: 978-0-9557300-8-5).
- [10] F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías. *Quality Issues on Model-Driven Web Engineering Methodologies*, Information Systems Development Springer Verlag, 2011, pp. 295–306, (ISBN: 10.1007/978-1-4419-7355-9_25).
- [11] F.J. Domínguez-Mayo, M. Espinilla, M.J. Escalona, M. Mejías, A Weighted Quality Evaluation Framework by applying the Analytic Hierarchy Process, *Nineteenth International Conference on Software Quality Management*, Global Quality Issues, 2011, pp. 55–66 (ISBN 978 1 907382 44 4).
- [12] F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías, M. Ross, G. Staples, A Quality Management Based on the Quality Model Life Cycle, *Computer Standards and Interfaces*, January 2012, doi: 10.1016/j.csi.2012.01.004 (ISSN 0920–5489).
- [13] Emasesa. <www.aguadesesevilla.com> (retrieved 04.11).
- [14] M.J. Escalona, G. Aragón, NDT. A model-driven approach for web requirements, *IEEE Transactions on Software Engineering* 34 (3) (2008) 377–390.
- [15] M.J. Escalona, N. Koch, Requirements engineering for web applications – a comparative study, *Journal of Web Engineering* 2 (3) (2004) 193–212.
- [16] Facebook for Developers. <<http://developers.facebook.com>> (retrieved 04.11).
- [17] I. Heitlager, T. Kuipers, J. Visser, A Practical Model for Measuring Maintainability, in: *6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, 2007.
- [18] M. Herrera, M.Á. Moraga, I. Caballero, C. Calero, Quality in Use Model for Web Portals (QiUWeP), in: *Proceedings of the 10th International Conference on Current Trends in Web Engineering*, Springer-Verlag, 2010, pp. 91–101.
- [19] D. Hoyle, *ISO 9000 Quality Systems Handbook – Updated for the ISO 9001:2008 Standard*, Using the standards as a framework for business improvement, Elsevier, 2009 (ISBN: 978-1-85617-684-2).
- [20] IEEE Std 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*.
- [21] ISO/IEC IS 9126, *Software Product Evaluation – Quality Characteristics and Guidelines for Their Use*, Geneva, Switzerland: International Organization for Standardization, 1991.
- [22] ISO/IEC 9126-1: *Software Engineering – Product Quality – Part 1: Quality Model*, Geneva, Switzerland: International Organization for Standardization, 2001.

- [23] ISO/IEC TR 9126-2: Software Engineering – Product Quality – Part 2: External Metrics, Geneva, Switzerland: International Organization for Standardization, 2003.
- [24] ISO/IEC TR 9126-3: Software Engineering – Product Quality – Part 3: Internal Metrics, Geneva, Switzerland: International Organization for Standardization, 2003.
- [25] ISO/IEC TR 9126-4: Software Engineering – Product Quality – Part 4: Quality in Use Metrics, Geneva, Switzerland: International Organization for Standardization, 2004.
- [26] ISO/IEC 25000:2005. Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE.
- [27] SC7, ISO/IEC DTR 25021: Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Quality Measure Elements ISO/IEC JTC1/SC7 WG6, January 11, 2006, 6N-565, 2006.
- [28] ISO/IEC 14598-1:1999. Information technology – Software product evaluation – Part 1: General overview.
- [29] ISO/IEC 15504-1:2004 Information technology – Process assessment – Part 1: Concepts and vocabulary.
- [30] ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.
- [31] S. Meliá, J. Gómez, S. Pérez, O. Díaz, A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA, in: ICWE '08 Proceedings of the 8th International Conference on Web, Engineering, 2008.
- [32] P. Mohagheghi, V. Dehlen, Developing a quality framework for model-driven engineering, Models in Software Engineering: Workshops and Symposia at MoDELS 2007 (2008) 275–286.
- [33] NDT (Navigational Development Techniques). <<http://www.iwt2.org/en/ndt.php>> (retrieved 03.11).
- [34] OMG: MDA. <<http://www.omg.org/mda/>> (retrieved 01.10).
- [35] L. Olsina, G. Lafuente, G. Rossi, Specifying Quality Characteristics and Attributes for Websites, Web Engineering (2001) 266–278.
- [36] L. Olsina, G. Rossi, Application Quality with WebQEM, 2002, pp. 20–29.
- [37] OOHDM (Oriented Hypermedia Design Method). <<http://www-di.inf.puc-rio.br/schwabe/HT96WWW/section1.html>> (retrieved 03.11).
- [38] R. Plösch, A. Mayr, C. Körner, Collecting Quality Requirements Using Quality Models and Goals, in: International Conference on the Quality of Information and Communications Technology, IEEE, 2010, pp. 198–203.
- [39] J.C. Preciado, M. Linaje, R. Morales-Chaparro, F. Sanchez-Figueroa, G. Zhang, C. Kroiß, N. Koch, Designing Rich Internet Applications Combining UWE and RUX-Method, in: ICWE '08 Proceedings of the 2008 8th International Conference on Web Engineering, 2008 (ISBN: 978-0-7695-3261-5).
- [40] T. Punter, R.O.B. Kusters, J.O.S. Trienekens, T. Bemelmans, A. Brombacher, The W-Process for Software Product Evaluation: A Method for Goal-Oriented Implementation of the ISO 14598 Standard, Review Literature And Arts Of The Americas, 2004, pp. 137–158.
- [41] T.L. Saaty, Introduction to a modeling of social decision process, Mathematics and Computers in Simulation 25 (1983) 105–107.
- [42] 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, G. Zhang, A survey on web modeling approaches for ubiquitous web applications, International Journal of Web Information Systems 4 (3) (2008) 234–305.
- [43] Twitter for developers. <<http://dev.twitter.com>> (retrieved 04.11).
- [44] UWE (UML-Based Web Engineering). <<http://uwe.pst.ifi.lmu.de>> (retrieved 03.11).
- [45] WebML (The Web Modeling Language). <<http://www.Webml.org>> (retrieved 03.11).