Qbox-Foundation: a Metadata Platform for Quality Measurement[1]

Lorena Etcheverry[†‡], Verónika Peralta[†‡], Mokrane Bouzeghoub[†]

† Laboratoire PRiSM, Université de Versailles
45, avenue des Etats-unis, 78035, Versailles Cedex, France
mok@prism.uvsq.fr
‡ Instituto de Computación, Universidad de la República
Julio Herrera y Reissig 565 5to piso, 11300, Montevideo, Uruguay
lorenae@fing.edu.uy, vperalta@fing.edu.uy

Abstract. Each application domain has its specific vision of data quality as well as a suite of (generally ad hoc) solutions to solve quality problems. However, there is an increasing interest in reusing quality knowledge and measurement methods. In this paper we present a metadata platform devoted to quality measurement. This platform is a foundation to a more complete toolset, named Qbox, defined in the Quadris project. Our platform is based on a quality metamodel which is a refinement of the Goal-Question-Metric and DWQ quality models. Specifically, this paper proposes (i) modeling general quality concepts and behaviors, (ii) implementing reusable measurement methods, and (iii) specializing concepts and methods for specific quality goals. The Qbox-Foundation provides an extensible collection of reusable measurement methods, supports their instantiation and automates their execution.

1 Introduction

Each application domain has its specific vision of data quality as well as a suite of (generally ad hoc) solutions to solve quality problems (Berty, 2004). However, there is increasing interest in reusing quality knowledge and measurement methods (Green, 2007) (Missier et al., 2003).

The quality of products and processes is traditionally assessed in a top-down way. The Goal-Question-Metric (GQM) paradigm proposes three abstraction levels: (i) at conceptual level, high-level quality goals are defined for products and processes, (ii) at operational level, a set of questions characterize the way to assess a specific goal, and (iii) at quantitative level, a set of quality measures is associated with each question in order to answer it. Information quality can also be analyzed under this paradigm; the DWQ quality model is an extended reuse of the GQM model in the context of data warehousing (Vassiliadis et al., 2000). In the context of the Quadris project, this latter model has been refined and adapted to a large class of applications (Akoka et al., 2007).

In this paper we present Qbox-Foundation, a metadata platform for quality assessment which aids in the definition of high-level quality goals and the specialization of typical

[1] This research was partially supported by the French Ministry of Research and New Technolologies under the ACI program devoted to Data Masses (ACI-MD), project Quadris.

measurement methods according to quality goals. Our main contributions are: (a) an improvement of the Quadris metamodel for understanding and reasoning with quality concepts, (b) an extensible collection of reusable quality metrics and measurement methods, (c) an interactive environment for instantiating quality metrics and measurement methods in order to fit specific goals and questions, and (d) a friendly interface for executing the specialized measurement methods and analyzing results.

Qbox-Foundation aims to provide generic concepts and processes which can be extended and refined to be adapted to specific quality decision applications. Although the definition of goals and questions is highly business-oriented and consequently it is not easy to reuse it in other application domains, the measurement phase is quite parametric and reusable metrics and measurement methods can be abstracted.

The specialization mechanism is based on an extensible catalog of quality metrics and parametric measurement methods. For example, a general purpose metric that measures *the amount of syntactic errors in a datum*, can be instantiated by specifying the types of syntactic errors to check for (which may be very different if we consider addresses, personal names or dates). Analogously, general purpose methods can be instantiated by setting appropriate parameters. Our proposal is based on three activities: (i) modeling general quality concepts and behaviors, (ii) implementing reusable parametric measurement methods, and (iii) specializing concepts and methods for specific quality goals. Qbox-Foundation already provides an extensible collection of quality concepts and reusable measurement methods. Then, quality analysts do not need to implement measurement methods but to instantiate them with the appropriate parameters. This considerably increases reuse in quality assessment applications.

The interactive environment of Qbox-Foundation aids business managers in the definition of quality goals, their decomposition in a set of questions and the association of questions with information system objects and quality concepts. Quality analysts also use this environment in order to instantiate quality metrics and methods. Once the assessment application is configured by instantiating all the appropriate methods, Qbox-Foundation runs measurement tasks and provides support to multidimensional analysis of the obtained measures. Specifically, Qbox-Foundation keeps histories of quality values, storing them in a multi-dimensional way, which allows the comparison of different assessment strategies, the discovery of quality trends, the exploration of interdependencies among quality dimensions and the management of quality evolution.

The remaining of the paper is organized as follows: Section 2 presents the quality assessment metamodel and Section 3 illustrates the instantiation mechanism for a case study. Section 4 describes Qbox-Foundation functionalities and provides implementation details. Finally, Section 5 presents our conclusions and future works.

2 Quality Assessment Metamodel

As mentioned before, our quality assessment metamodel is a result of successive refinements of the Goal-Question-Metric (GQM) paradigm (Basili et al., 1994), done in DWQ (Vassiliadis et al., 2000) and Quadris (Akoka et al., 2007) projects. Figure 1 gives a synthesized picture of this metamodel.

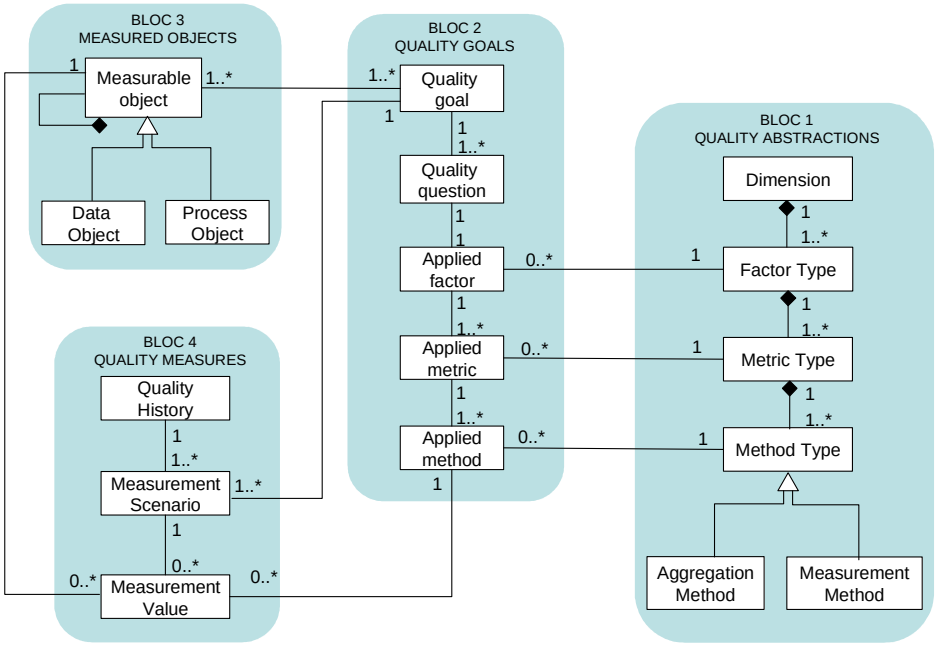

Fig. 1 – *Quality assessment metamodel*

The first bloc of this quality metamodel constitutes a library of abstract data types which will be used to characterize specific quality goals. The main abstractions of this part of the metamodel are:

- *Quality dimensions*: Traditionally, information quality is characterized via multiple dimensions, which help to rank data (e.g. freshness, accuracy, completeness) or the processes that manipulate this data (e.g. response time, reliability, security). A dimension captures a high-level facet of quality.
- *Quality factors*: A factor represents a particular aspect of a quality dimension, for example, data accuracy involves semantic correctness, syntactic correctness and precision of data (Peralta, 2006). There might be several factors for the same dimension; each factor best suites a particular problem or type of system.
- *Quality metrics*: A metric is an instrument used to measure a certain quality factor, for example the percentage of system data that match real-world data is a metric for semantic correctness. There might be several metrics for the same quality factor.
- *Quality methods*: A method is a process that implements a quality metric. Two types of methods are defined: (i) *measurement methods*, which compute the quality of an object by directly measuring it (e.g. counting the number of null values in a tuple), and (ii) *aggregation methods*, which compute the quality of a composed object by aggregating quality values of object parts (e.g. computing precision of a table by averaging the precision of its tuples). There might be several methods to implement the same metric.

This library of abstractions is extensible, in the sense that new concepts can be added in order to manage more quality aspects. In addition, the library is general enough to manage different application domains. In order to adapt quality concepts to specific application scenarios, we need to instantiate them taking into account the particularities of specific quality questions. First of all, quality factors may be specialized in order to best suit a quality question (e.g. syntactic correctness *of addresses*). Then, quality metrics and methods of such factor may be specialized in order to access the corresponding IS object (e.g. checking for specific syntactic errors that commonly appear in address data). The *Applied factor*, *Applied metric* and *Applied method* classes represent instantiated quality concepts.

The second bloc of the metamodel deals with quality goals. More specifically, it represents the GQM approach with a specific refinement of the metric level considering the abstraction introduced in the previous bloc. However, we still consider the model defined at three levels:

- Goal level: A goal represents a high-level quality need. An example of a goal may be “reducing the number of returns in customer mails”. Goals are related to specific business objects (e.g. customers) in a particular environment (e.g. mail delivery) or business process (e.g. improve application performance). Complex goals may be decomposed into subgoals.
- Question level: A question represents the ultimate refinement/decomposition of a goal or subgoal. A refinement corresponds to a question if the corresponding quality assessment can be characterized by a unique quality factor. The set of questions and their corresponding quality factors, related to a specific quality goal, implement the way this goal should be performed. Goal questions fix the objects subject to measurement (e.g. customer addresses) with respect to a selected quality aspect (e.g. syntactic correctness) and determine their quality from the selected viewpoint (e.g. marketing manager). An example of question associated to the previous goal may be “reducing syntactic errors in customer addresses”.
- Metric level: In our approach, this level is actually refined into three sublevels, associated to the hierarchy of abstraction given in the first bloc of the metamodel: quality factor sublevel, quality metric sublevel and quality value sublevel. Given a quality question, the answer to this question is defined by choosing a quality factor which best characterizes the question, a metric which is appropriate to measure this factor and a method of measurement of this metric.

These three levels allow specifying a quality goal with respect to two dimensions: the generic quality concepts (bloc 1 of the metamodel) and the information system object types (bloc 3 of the metamodel).

The third bloc of the metamodel refers to the information system model and to the processes which operate on the instances of this model. Each object type, being either a data or a process, is called a measurable (or measured) object if it is subject to a qualitative evaluation within a quality goal. The details of the information system model and processes are out of the scope of this paper.

The fourth bloc of the metamodel deals with measurements. Given the definition of a quality goal, at any moment there will be a need to evaluate the quality questions and to analyze the obtained values in the perspective to improve the quality of the measured objects. Each goal measurement is called a *measurement scenario* and is composed of the set of values respectively associated to the set of questions defining the quality goal. Results of

successive quality scenarios is called a *quality history*, it serves to analyze behaviors and trends of the measured objects. Generally, improvement actions are taken based on this analysis. Improvement actions definition is out of the scope of this paper.

3 Instantiation of the Metamodel with a Case Study

In this section we show the usage-aspects of Qbox-Foundation following a simple academic case study. The analyzed application corresponds to an information system that handles information about students at a university (Etcheverry et al., 2007). We distinguish 4 different actors using Qbox-Foundation:

- *Quality management experts*: Responsible for the definition and maintenance of the library of quality concepts (bloc 1 of the quality metamodel),
- *Business manager*: Responsible for the definition of quality goals and questions as well as their association with quality factors and IS object types (first part of bloc 2)
- *IS administrator*: Responsible for assuring the access to IS objects (bloc 3),
- *Quality analyst*: Responsible for the specialization of metrics and methods, the execution of methods and the analysis of results (alerts, trends, etc.) (last part of bloc 2 and bloc 4).

In order to help quality management experts, we have implemented an initial library of quality methods. Table 2 lists some examples of methods, corresponding to data accuracy metrics. Definitions of the accuracy dimension, its factors and metrics have been taken from (Peralta, 2006); they are summarized in Table 1.

Accuracy: It is concerned with the correctness and precision with which real world data of interest to an application domain is represented in an information system	
Semantic correctness	It describes how well data represent states of the real-world
Semantic correctness Boolean	A Boolean indicating whether a system datum corresponds to real-world
Semantic correctness degree	A degree indicating the impression/confidence on whether a system datum corresponds to real-world
Semantic correctness deviation	The semantic distance between a system datum and its correspondent datum in real-world
Syntactic correctness	It expresses the degree to which data is free of syntactic errors such as misspellings and format discordances
Syntactic correctness Boolean	A Boolean indicating whether a system datum satisfies syntactical rules
Syntactci correctness deviation	The syntactic distance between a system datum and a reference one considered as syntactically correct
Precision	It concerns the level of detail of data representation
Scale	The precision associated to the measurement scale
Standard error	The standard deviation of a set of measurements
Granularity	The number of attributes used to represent a single concept

Tab. 1 – *Accuracy factors and metrics*

Method (and metric)	Description	Parameters
CheckReferential (sem. corr. Boolean)	Checks if a given datum corresponds to an entity (given its key) by looking in a referential.	-<key, attribute> to check -Referential table -Comparison function (equality, similarity, …)
CheckRule (synt. corr. Boolean)	Checks if a given datum satisfies a format rule.	-Attribute to check -Format rule
CheckDictionary (synt. corr. Boolean)	Checks if a given datum is present in a dictionary.	-Attribute to check -Dictionary
ComputeDistance (synt. corr. deviation)	Computes the distance between a given datum and the most similar datum contained in a dictionary.	-Attribute to check -Dictionary -Distance function
ComputePrecisionLevel (granularity)	Returns a precision level (in certain scale) according to the number of null values of an entity.	-Set of attributes to check -Precision scale

Tab. 2 – *Some measurement methods for accuracy metrics*

Business managers define quality goals and decompose them into a set of quality questions, setting the concerned IS objects and the associated quality factors. Table 3 illustrates the decomposition of a given goal into a set of questions and their association with IS objects and quality factors. Quality factors are selected from the library of factor types and possibly renamed or adapted (e.g. changing description) in order to better fit the question.

Goal: Improve the quality of students location data (phone number, address, etc)			
Question		**IS objects**	**Quality factor**
1	Are students' addresses the correct ones?	Student's address	Sem. corr.
2	Are the students' addresses correctly written?	Student's address	Synt. corr.
3	Are the students' telephones valid ones?	Student's telephone	Synt. corr.
4	Do we have precise students' addresses?	Student's address	Precision
5	Are students' addresses up to date?	Student's address	Currency
6	Do we have all students' addresses?	Student's address	Coverage

Tab. 3 – *Decomposition of a quality goal and association with IS objects and a quality factor*

For each quality question, a *quality analyst*, who should have a good understanding of the application domain, the underlying IS and the quality library, chooses appropriate metrics and methods and instantiates them to the quality question. For metrics, instantiation consists in selecting a metric type and (possibly) adapting its name, description and units in order to better fit the quality question. For methods, instantiation consists in choosing a method type and setting its parameters (e.g. set the format rule of the CheckRule method). If the analyst doesn't find any suitable method type in the library, he may define a new method (possibly modifying and existing one) and add it to the library. Table 4 shows some examples of applied metrics and methods for some of the questions of Table 3.

Question	Metric	Method	Instantiated parameters
1	Address sem. corr. Boolean	CheckReferential	<student's id, student's address>; university administrative DB; equality
2	Address synt. corr. Boolean	CheckDictionary	student's street; street dictionary
2	Address synt. corr. deviation	ComputeDistance	student's street; street dictionary; string-edit-distance
2	Address synt. corr. Boolean	CheckRule	student's address; {street standard format}
4	Address granularity	ComputePrecision Level	{student's street, door number and city}; {1 if none is null, 0.8 if only door number is null…}

Tab. 4 – *Instantiation of metrics and methods for some quality questions*

The instantiation of factors and metrics (renaming and adapting descriptions) facilitate the search of similar factors/metrics and their reuse for new questions. For example, the factor of question 2 (see Table 3) may be called *address syntactical correctness*. Later, somebody needing quality metrics and methods in order to analyze teacher's addresses may reuse it, and possibly refine its metrics and methods. An already instantiated method (e.g. CheckRule) may be directly used or may be further specialized defining a new method (e.g. changing the format rule in order to include affiliation information in addresses of external teachers). Furthermore, success stories of other application domains can be adapted for specific applications.

4 Qbox-Foundation Design and Implementation

Qbox-Foundation was implemented as a Java web application, with user interfaces for managing the different entities of the metamodel and executing measurement methods. Its main functionalities include:

- Management of an extensible library of dimension, factor, metric and method types. There are methods for retrieving and editing concepts and incorporating new ones. We have chosen a tree-like structure to show this information to the user (see bottom panel of Figure 2). We provide an interface for developing new methods (descriptions and code) or defining methods that invoke external routines.
- Definition and storage of user's quality goals and questions. We provide methods for defining and editing quality goals and decomposing them into quality questions. A drag-and-drop interface allows browsing among IS objects and associating them with questions. This association allows tracking the influence of IS objects quality with respect to specific questions. Analogously, quality factors can be instantiated and associated to questions in a drag-and-drop way. This interface (Figure 2) is the starting point for configuring a new quality-assessment application in the Qbox-Foundation.
- Association of quality metrics and measurement methods with quality questions. The configuration of a quality assessment-application finishes by choosing the appropriate metrics and methods and instantiating them according to the question. Is

in this step when the quality analyst actually determines what is going to be measured. To this end, a drag-and-drop interface facilitates the browsing among the library of quality concepts and the parameterization of methods. New metrics and methods can be easily defined, either by modifying existing ones or by defining them from scratch.

- Execution of measurement methods for individual IS objects (or all objects) involved in a given quality goal, and persistency management of the obtained quality values. Specifically, Qbox-Foundation keeps histories of quality values.
- Show results, allowing the visualization of trends and correlations. Quality values are stored in a multi-dimensional way, which allows the comparison of different assessment strategies, the discovery of quality trends and the exploration of interdependencies among quality factors. The storage of historical values also allows exploring which measurement methods are best suited for each situation and managing quality evolution.

The following screenshot illustrates the Qbox-Foundation interface (see Figure 2). The tree in the upper left corner shows the defined goals and questions (those of Table 3), and for each question the associated quality factor. The tree in the upper right corner allows browsing among IS objects (in this example the database that represents students). Finally, the tree in the lower part of the screen shows the library of quality concepts, allowing browsing and choosing appropriate factors, metrics and methods.

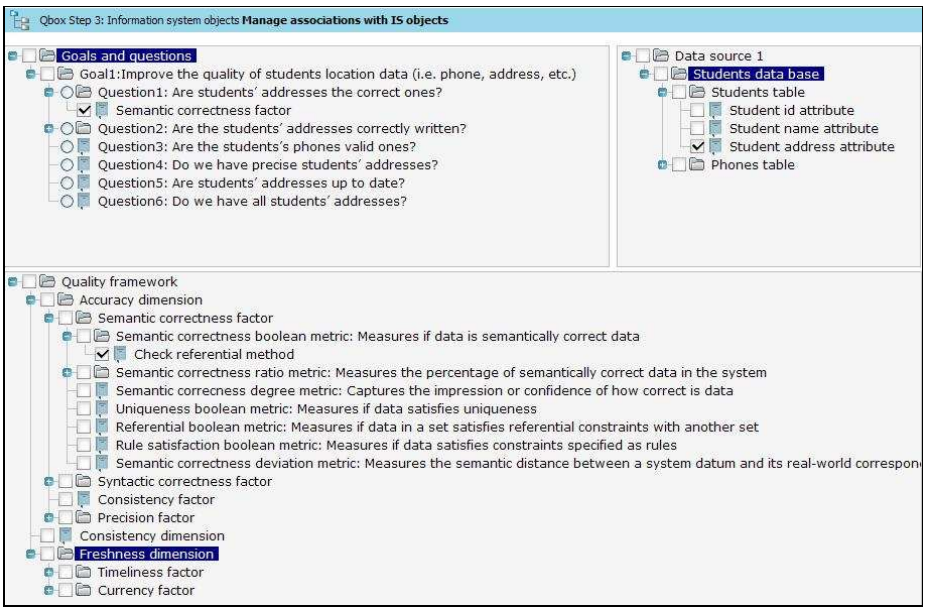

Fig. 2 – *Qbox-Foundation interface*

The implementation of Qbox-Foundation is based on Struts framework and uses JPivot and Mondrian for analysis of results. Deployment was carried out with a Tomcat JSP container, a Mondrian OLAP server and a PostgreSQL DBMS.

Figure 3 shows the architecture of the tool. The Data Access Layer encapsulates the access to IS objects and implements persistence mechanisms over the Qbox-Foundation Respository. The Logic layer contains the implementation of the measurement methods and the analysis component. The Presentation Layer is implemented as jsp files and uses the JPivot component in order to show the measurement results.

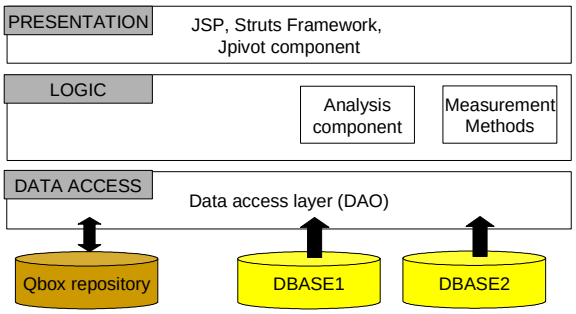

Fig. 3 – *Qbox-Foundation architecture*

5 Conclusions

In this paper, we presented the Qbox-Foundation which is a platform devoted to quality management of information systems. The Qbox-Foundation is the basement of the Qbox toolkit proposed in the Quadris project in order to support quality applications development and to handle multiple quality factors analysis. The Qbox-Foundation implements a quality metamodel and a library of measurements methods and offers multiple operations for executing these methods, achieving the derived values and providing multidimensional support for organizing and browsing these values. The metamodel supported by the Qbox-Foundation is a refinement of the Quadris metamodel presented in (Akoka et al., 2007). Further work will focus on the multidimensional analysis and on studying correlations between quality factors through measurements obtained from real application datasets. The ultimate goal is to derive from this study a collection of quality patterns which can be used for quality assessment of different application domains.

References

Akoka, J., L. Berti-Equille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdoué-Thion, Z. Kedad, S. Nugier, V. Peralta and S. Sisaid-Cherfi (2007). A Framework for Quality Evaluation in Data Integration Systems. *9th International Conference on Enterprise Information Systems (ICEIS'2007)*, Funchal, Portugal.

Basili, V., G. Caldiera and H.D. Rombach (1994). The Goal Question Metric Approach. *Encyclopedia of Software Engineering*, 528-532, John Wiley & Sons, Inc.

Berty-Equille, L. (2004). Un etat de l'art sur la qualité des données. *Ingénierie des systèmes d'information*, 9(5-6):117-143.

Etcheverry, L., S. Tercia, A. Marotta and V. Peralta (2007). Medición de la exactitud de datos en sistemas fuentes: un caso de estudio. Technical report, Universidad de la República, Uruguay

Green, B. (2007) Information Management Standards and Data Quality Thematic Briefing Paper (May 2007). Europe's one-stop shop on Public Sector Information re-use. URL: www.epsiplus.net; accessed on December 2007.

Missier, P., G. Lalk, V. Verykios, F. Grillo, T. Lorusso and P. Angeletti (2003). Improving Data Quality in Practice: A Case Study in the Italian Public Administration. *Distributed and Parallel Databases*, 13 (2).

Peralta, V. (2006). *Data Quality Evaluation in Data Integration Systems*. PhD thesis, Université de Versailles, France & Universidad de la República, Uruguay.

Scannapieco, M., P. Missier and C. Batini (2005). Data Quality at a Glance. *Datenbank-Spektrum*, 14: 6-14.

Vassiliadis, P., M. Bouzeghoub and C. Quix (2000): Towards Quality-oriented Data Warehouse Usage and Evolution. *Information Systems*, 25(2): 89-115.

Résumé

Chaque domaine d'application a des visions spécifiques de la qualité de l'information ainsi que des batteries de méthodes (généralement ad hoc) pour résoudre des problèmes de qualité. Cependant, les organisations ont un intérêt croissant pour la réutilisation des techniques et des méthodes de mesure de la qualité. Dans cet article, nous présentons une plateforme de méta données dédiée à la mesure de la qualité. Cette plateforme est une fondation pour une boite à outil plus complexe, nommée Qbox, définie dans le projet Quadris. Notre plateforme est basée sur un méta modèle de qualité, qui est un affinage des modèles de qualité de GQM (Goal-Question-Metric) et de DWQ (Data Warehouse Quality). En particulier, nous proposons de : (1) modéliser les concepts généraux de la qualité, (2) implémenter des méthodes de mesure réutilisables et (3) spécialiser les concepts et les méthodes par rapport à des buts de qualité spécifiques. Qbox-Foundation fournit une collection extensible de méthodes de mesures réutilisables, supporte leur instanciation et automatise leur exécution.