

Towards a software quality certification of master data-based applications

Fernando Gualo^{1,2}  · Ismael Caballero^{1,2}  · Moisés Rodríguez^{1,2,3} 

Published online: 20 January 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Master data management (MDM) can provide an integrated and unified view of key business entities to offer better support in business processes. Due to the very nature of master data-based applications, it is possible to use data with the highest possible level of quality. MDM can help ensure that some common concerns, like duplicates or inconsistencies, are prevented by sharing a ‘single version of the truth’ throughout the organisation, and, in some cases, allowing collaborative updates to the master data repository. Therefore, assuring the reliability of master data-based applications, would improve the organisation efficiency. This type of application should implement a set of functional requirements covering the basic operation of MDM principles. We propose a solution based on the evaluation and certification of ‘functional suitability’ of MDM applications. As part of our proposal, we inferred a set of functional requirements from parts 100 to 140 of ISO 8000. This set will be used as a reference in the required matching to compute values for each one of the metrics, properties, subcharacteristics and ultimately, functional suitability following a bottom-up procedure. Finally, the paper also describes the application of the evaluation procedure of an existing master data-based application.

Keywords Master data management · Certification · Data quality management · ISO 8000 · ISO 25010 · Functional suitability

✉ Fernando Gualo
fgualo@dqteam.es

Ismael Caballero
Caballero@uclm.es

Moisés Rodríguez
mrodriguez@aqclab.es

¹ DQTeam SL, Paseo de Moledores s/n, 13071 Ciudad Real, Spain

² Instituto de Tecnologías y Sistemas de Información, UCLM, Paseo de Moledores s/n, 13071 Ciudad Real, Spain

³ AQCLab, Paseo de Moledores s/n, 13071 Ciudad Real, Spain

1 Introduction

Master data describes the essential business entities of a company, such as ‘suppliers’, ‘customers’, ‘products’, ‘employees’ or ‘assets’—for example—(Spruit and Pietzka 2014; Smith and McKeen 2008; Dreibelbis et al. 2008; Loshin 2010), and represents the business objects that are shared across transactional applications. Master data enables the creation of a ‘single version of the truth’ about these objects across the operational IT landscape (Allen and Cervo 2015). The instances or values of the master data classes do not relatively change over time, especially, compared to transactional data. Reference data that represents master data can be used for capturing transactional data in operations (Otto et al. 2012; Mahanti 2019). Master data-based applications (hereafter, MDM-based applications) should be able to manage master data and reference values during their underlying operations.

Master Data Management Server (MDM server) is a software product built on top of a database technology—typically relational—for backend processing. They usually provide service-oriented architecture endpoints to provide master data instances to Master data-based applications (Dreibelbis et al. 2008). The communication between an MDM server and MDM-based application is typically conducted through the exchange of master data messages. Several examples of master data message types are covered in Rivas et al. (2017).

Hence, it would be desirable that the organisation’s MDM-based applications are reliable for all the stated uses. In addition to this, having available trustable master data directly benefits the quality of the data used by the business processes (Cleven and Wortmann 2010). It brings some advantages, such as the reduction or complete elimination of duplicate records in critical datasets; the availability of current, cleansed, consistent master data throughout the enterprise; the entity resolution procedure (Talburtt 2011); the automatisisation of probabilistic matching and application of business rules; and a potential collaborative management of the master dataset (Fleckenstein and Fellows 2018).

To make MDM-based applications more trustable and reliable, we pose that a specific certification from ISO 8000-1x0 compliance would help to achieve this goal. From our point of view, this certification would be achieved through the evaluation of the software quality characteristic of ‘functional suitability’. This implies to define a set of functional software requirements, which should be used as a reference during the evaluation. After analysing the current state of that type of software quality certification (see Sect. 5), and to the best of our knowledge, we can conclude that the following two problems need to be addressed:

1. There is no generic environment for the evaluation and certification of the master data-based systems
2. There are no standards specifically addressing any set of functional software requirements that should be used as a reference.

To address the first problem, we propose the use of ISO 25010 (ISO 2011a) and ISO 25040 (ISO 2011b) for the evaluation of software product based on functional suitability—as it is specified in the certification environment proposed by Rodriguez et al. (2016) (see Appendix).

With regard to the existence of international standards covering the functional software requirements of master-based systems, we found that only the parts ISO 8000 parts 100 to 140 (ISO 2009, 2016a, b, c, d) cover specifically this area. Consequently, we decided to use this family of standards to infer a set of functional software requirements to be used as a reference for the certification of ‘functional suitability’.

Hence, the main contribution of this paper is twofold:

1. The identification of a set of functional requirements inferred from ISO 8000 that any master data-based application should meet
2. The application of the complete environment defined by Rodriguez et al. specifically customised with the set of the identified functional requirement to a specific case study of a master data-based application

It is important to highlight that the scope of the certification we are proposing will only be conducted considering the requirements that we inferred from the ISO 8000 study, parts 100 to 140. Additionally, it is possible that the evaluated systems could potentially implement some other different software requirements, which are out of the scope of this certification.

The remainder of the paper is structured as follows: Sects. 2 and 3 briefly introduce the related works in MDM-based application certification and the main concerns that parts 100 to 140 of ISO 8000 present. Section 4 presents the set of software requirements inferred from ISO 8000 parts 100–140. Section 5 describes the application of the evaluation process of the functional suitability to an existing master data-based application through a case study. Finally, Sect. 6 presents some conclusions.

2 Related works

To the best of our knowledge, we can conclude that there are no researches related to the evaluation of master data-based application about ISO standards focused on master data management. We have found several works related to quality in master data.

The first proposal is focused on the certification of software products in relation to ISO 8000 part 110. ECCMA¹ is an American association dedicated to the creation of technical data dictionaries according to specific domains. ECCMA developed the ECCMA Open Technical Dictionary (eOTD), and they are offering its certification exclusively for customers who work with its eOTD data dictionary. ECCMA certifies and registers a specific version and release of a software application to be compliant with ISO 8000-110:2009. It is conducted by demonstrating that the application can access the eOTD using web services and import and export Identification Guides. However, this ECMMA certification only covers the requirements of ISO 8000-110, but not the requirements from ISO 8000, parts 120 to 140.

The second proposal is focused on maturity models for master data management. Different maturity models to master data management are introduced in (Spruit and Pietzka 2014; Otto et al. 2012; Choi et al. 2010; Zúñiga et al. 2018). These maturity models allow to determine the degree of maturity at which the company is ready to implement master data management solutions, and/or the degree to which these solutions are implemented. In (Spruit and Pietzka 2014) a maturity model is proposed to determine whether the organisation is prepared to implement a master data solution. In (Otto et al. 2012), the design process toward a functional reference model for master data quality management (MDQM) is described. This model design process spanned several iterations comprising multiple designs and evaluation cycles. Choi et al. (2010) suggests an enterprise master data model (EMDM), which defines all of the master data composition elements, the hierarchical structure in each master data element, and

¹ <https://www.eccma.org/certification/#iso3>

the relationship among master data elements across the organisation. Finally, in (Zúñiga et al. 2018), a maturity model for microfinance sector and a study case in this industry is presented. All these works are focused on the maturity of the organisation to adopt a master data-based application and its capability, but none of these is focused on these applications.

Finally, Otto et al. (2010) proposes a method to measure the quality of master data in the business and how this quality affects business decisions and strategies. This proposal is focused on the quality of the master data utilised in the organisation. Notwithstanding, this proposal is aimed neither at the MDM-based application nor the requirements that this type of application must meet.

All things considered, and based on our research, it can be concluded that there are works in master data and its quality, but they are solely focused on the capability of the organisation to implement master data-based application or in the master data itself. Furthermore, ECCMA certifies master data-based applications but it only considers the requirements of ISO 8000 part 100, not the functional requirements from parts 120, 130 and 140, which are included in the scope of this proposal. So, the main advantage of our proposal is its wider coverage than ECCMA's one.

3 General overview of ISO 8000-parts 100 to 140: exchange of characteristic data in master data

Parts 100 to 140 of the ISO 8000-1x0 family of standards describe some aspects of the exchange of master data between applications (or organisations) using the exchange of master data messages. To this effect, the family of standards proposes the usage of a pre-set template to construct the messages and to encapsulate the master data within the messages. Similarly, the requirements that these messages must meet in order to incorporate information on the level of provenance, accuracy and completeness of the data contained in the master data message, are specified. Moreover, it is important to know other elements included and related in Fig. 1.

Each part of the standard covers one of the aspects mentioned above. The contents of each part of the family are summarised below:

- ISO 8000-100: 'Master data: Exchange of characteristic data: Overview' (ISO 2016a) offers an outline of the exchange of master data.
- ISO 8000-110: 'Master data: Exchange of characteristic data: Syntax, semantic encoding, and conformance to data specification' (ISO 2009) include the requirements to build and exchange master data messages: adherence to formal syntax, semantic encoding, compliance of a customer's data specification, business model.
- ISO 8000-115: 'Master data: Exchange of quality identifiers: Syntactic, semantic, and resolution requirements' (ISO 2018a) indicates the syntactic and semantic requirements for the construction and resolution of quality identifiers in the exchange of master data messages.
- ISO 8000-116: 'Master data: Exchange of quality identifiers: Application of ISO 8000-115 to authoritative legal entity identifiers' (ISO 2018b) indicates the format of legal entity identifiers (ALEI) for individuals and organisations.
- ISO 8000-120: 'Master data: Exchange of characteristic data: Provenance' (ISO 2016b) introduces a data model to describe the information about the origin of the data. This part

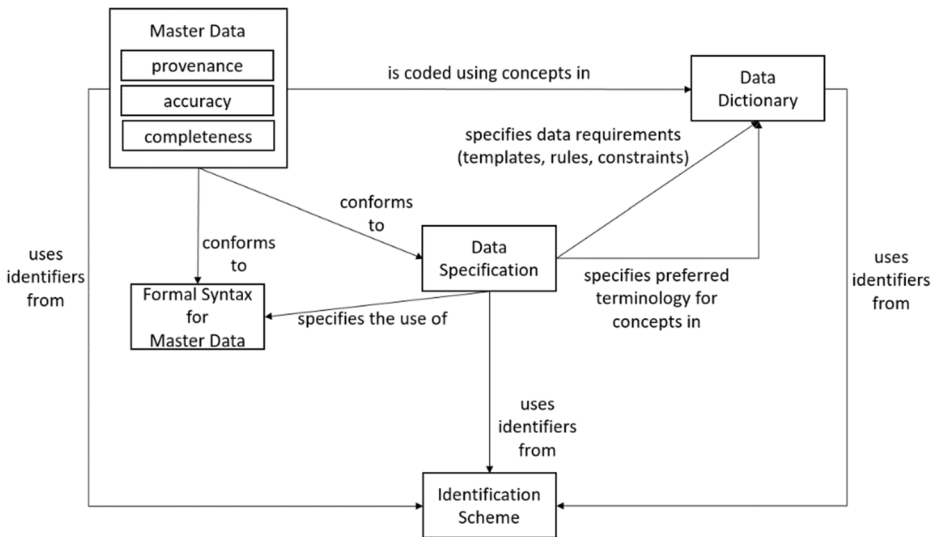


Fig. 1 Data architecture for master data management (adapted from (ISO 2016a))

of the standard looks at the most recent owner of the data. Hence, it is focused on two aspects: When the data was extracted from the database, and who the owner of the repository is.

- ISO 8000-130: ‘Master data: Exchange of characteristic data: Accuracy’ (ISO 2016c) describes how to add information about the accuracy of the exchanged master data.
- ISO 8000-140: ‘Master data: Exchange of characteristic data: Completeness’ (ISO 2016d) describes how to add information about the completeness of the exchanged master data.
- ISO 8000-150: ‘Master data: Quality management framework’ (ISO 2011c) specifies the fundamental activities of the management of the quality of the master data.

4 Set of functional software requirements for master data-based application inferred from ISO 8000

Let us remind that our approach is grounded on the certification of the software quality characteristic of ‘functional suitability’, as described in ISO 25010, and following the model introduced by [RefMoises]. To enable the certification, a set of functional requirements is required as a reference to measure the characteristics, subcharacteristics and properties of the functional suitability. As part of our investigation, we developed this reference set of functional requirements that any master data-based system must implement.

This section presents the set of functional requirements that we infer from the study of ISO 8000, parts 100 to 140. As far as we are concerned, there is currently no international open standard (such as the ones developed by ISO or IEEE) that specifically address the functional software requirements that MDM-based applications must have; we have only found a family of standards for the exchange of characteristic data in master data (see Sect. 3). We did not consider the inclusion of ISO 8000-150 because this part does not specifically address any concern related to requirements which can be implemented in master data-based systems. Due to the same reason, we did not consider any of the ISO 8000 parts 60, 61 or 62.

In general terms, we will assume that the master data exchange functionalities are a subset of the functionalities that a whole MDM-based application should meet. Therefore, we will take this family of standards as the main input to identify and list the functional software requirements that will be used as a reference for the evaluation of the functional suitability of this type of application.

We will establish the normative clauses of ISO 8000-1x0 standards, those that identify possible mandatory functional requirements that any MDM-based applications should meet. These requirements are grouped into three categories: those related to master data messages construction and use, those related to data dictionary management and those related to the exchange of master data messages.

In the following subsection, every group of requirements will be described.

4.1 Functional requirements related to master data message construction and use

This subset is focused on the creation and validation of the master data messages, and the CRUD operations to access the master data repositories managed by the Master Data Server via Master Data Message exchange.

1. *FR.1. Build master data messages* [inferred from ISO 8000-100, clause 6]. MDM-based application must be able to construct master data messages according to a formal syntax defined in an information model. This formal syntax must be known by all parties in the organisation, and by all organisations that will exchange master data between them.
2. *FR.2. Code and/or decode master data message and records* [inferred from ISO 8000-110, clause 8]. MDM-based application must be able to encode the identifiers and property values of each of the data dictionary entries in the construction of the master data message. In addition to this, the MDM-based application must be able to encode the complete master data message before the exchange. Finally, the MDM-based application must be able to decode the master data message and the property values in the MDM system receiving the message.
3. *FR.3. Verify correctness of master data message* [inferred from ISO 8000-110, clause 6 and 7]. MDM-based application must verify the correctness of the master data message against its formal syntax and its data specification automatically in computer-based way. The whole organisation and the MDM-based application must know previously the formal syntax.
4. *FR.4. Record messages of provenance-related events* [inferred from ISO 8000-120, clause 6.5]. MDM-based application must allow keeping records of origin. Each one of the provenance-related events must contain a unique and univocal identifier, the provenance-related event, and an assertion of the event.
5. *FR.5. Record provenance-related events in the records* [inferred from ISO 8000-120, clause 7]. MDM-based application must allow storing provenance-related events. For each provenance-related event, it is necessary to know the type of event, the organisation that it refers to, the person who records the event and his/her role in the organisation, and a timestamp of the provenance event. All this information must be included for each attribute in the master data message.
6. *FR.6. Record accuracy-related events in the records* [inferred from ISO 8000-130, clause 6.3, 7 and 9]. MDM-based application must record all the accuracy-related events. For each accuracy-related event, it is necessary to know details about the event, the type or

kind of event, the accuracy method used in the message, the organisation that it refers to, the person who records the event and his/her role in the organisation, and a timestamp of the accuracy event.

7. *FR.7. Validate accuracy-related events through an accuracy method* [inferred from ISO 8000-130, clause 6.4]. Validate accuracy-related events through an accuracy method. MDM-based application should have several standardised accuracy methods that must be used to validate the accuracy message and the content of this message. These methods must be known throughout the organisation.
8. *FR.8. Record completeness-related events in the records* [inferred from ISO 8000-140, clause 6.3, 7 and 9]. MDM-based application must record completeness-related events. For each completeness-related event, it is necessary to know details about the event, the type or kind of event, the completeness method used in the message, the organisation that it refers to, the person who records the event and his/her role in the organisation, and a timestamp of the completeness-related event.
9. *FR.9. Validate completeness-related events through a method of completeness* [inferred from ISO 8000-140, clause 6.4]. Validate completeness-related events through a method of completeness. The MDM-based application has several standardised completeness methods that must be used to validate the accuracy message and the content of this message. These methods must be known throughout the organisation.

4.2 Functional requirements related to data dictionary management

The set of functional requirements below is focused on the connection, management, and CRUD operation into the data dictionary.

10. *FR.10. Query the data dictionary* [inferred from ISO 8000-100, clauses 6 and 8]. MDM-based application must access the data dictionary in order to get an entry from the data dictionary before the build and the exchange of a master data message.
11. *FR.11. Create a new property in the data dictionary* [inferred from ISO 8000-110, clause 8.2.]. MDM-based application must be able to create new properties (attributes) for an entity in the data dictionary.
12. *FR.12. Assign a value to a property* [inferred from ISO 8000-110, clause 8.2.]. MDM-based application must be able to assign a value 'v' to a property 'p' in a master data record of the data dictionary.

4.3 Functional requirements related to the exchange of master data messages

The set of requirements below are focused on the exchange of master data messages.

13. *FR.13. Exchange master data message* [inferred from ISO 8000-120, clause 1, ISO 8000-130, clause 1 and ISO 8000-140, clause 1]. The MDM-based application must be able to exchange master data messages to supply or to acquire data. The MDM-based application must provide a communication layer composed of web services in order to exchange (send and receive) master data messages

The presented set of functional requirements covers all type of operations that any master data-based systems should do. Consequently, we consider that the presented set is both comprehensive and complete.

5 Case study

In this section, a case study is introduced. It presents the evaluation of a real MDM-based application according to the set of functional software requirements explained in Sect. 3. The case study is composed of two evaluations. The first evaluation is the procedure over the first version of the MDM-based application. This first version of FCD does not implements all the functional requirements from the set of functional software requirements inferred from ISO 8000 parts 110 to 140. Therefore, in this first evaluation it was obtained a low level of quality for functional suitability; and a second evaluation conducted on an improved version of the MDM-based application, which implements a set of improvement actions over the first version of the MDM-based application. Hence, in the second evaluation, it was obtained a higher level of quality for functional suitability. The quality level of functional suitability is represented on a level scale which ranges from 1 to 5, where 1 is the lowest level and 5 is the highest. For both evaluations, the evaluation process used was based on ISO 25040 (ISO 2011b) presented in Appendix.

5.1 The evaluated product

We have evaluated, as a case study, the software ‘Data Quality Firewall’ (FCD for its Spanish acronym ‘Firewall Calidad de Datos’), whose architecture was firstly presented in (Rivas et al. 2017) and shown in Fig. 2. FCD is an MDM-based application used in a consultancy multinational company (anonymised and hereafter called ‘OX’) with more than 23,000

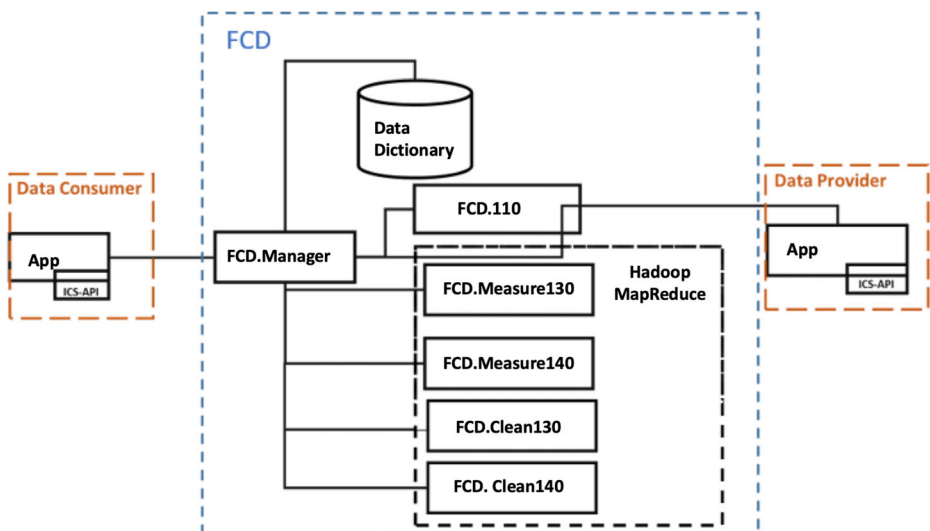


Fig. 2 FCD logical design

workers all over the world. FCD is composed of different software artefacts providing the following functionalities:

- Facilitate the exchange of master data among organisations through the mapping of vocabularies used by different organisations representing the same concepts to facilitate data integration of the data.
- Measure and add information about the level of accuracy of the data exchanged among organisations. This involves the accuracy levels evaluation of the data being exchanged.
- Measure and add information about the completeness level of the data exchanged. This involves the completeness evaluation of the data being exchanged.
- Perform data cleansing operations to curate the master data being exchanged.

All the abovementioned functionalities enable the exchange of master data from two applications (data provider and data consumer) through FCD ensuring that the master data has due levels of quality.

5.2 First evaluation of the FCD

In this section, the main results of the application of the evaluation process are presented, based on ISO 25040 (ISO 2011b) introduced in Appendix to FCD, considering the necessary set of functional software requirements presented in Sect. 3. Therefore, the most interesting topics will be described for each activity.

As a result of the execution of the *Activity 1. Establishing the evaluation requirements*, the quality model was presented to the customer by the evaluation team, as well as the explanation of the inferred functional requirements. The customer provided a list of the whole set of functional requirements implemented in FCD. This activity is performed during several meetings. Since the implementation of the FCD could not explicitly cover the reference set of the functional requirements for the certification, it was necessary to analyse the whole set of functional requirements of FCD to identify those susceptible to be mapped against the *reference set of requirements—inferred from ISO 8000 parts 100 to 140*—in order to be included in the evaluation. As a result, 16 out of more than 60 requirements were identified (see Table 1).

Once identified the requirement candidate to be included in the evaluation, it is necessary to establish a relationship to fully determine whether FCD implements all the requirements set of

Table 1 FCD's functional requirements candidates to be studied as part of the certification

FCD functional requirements	
FR.1. Create provenance event	FR.9 Delete vocabulary
FR.2. Create accuracy event	FR.10 List vocabularies
FR.3. Create completeness event	FR.11. Add term
FR.4. Create master data message	FR.12. Modify term
FR.5. Add header to event	FR.13. Delete term
FR.6. Add data to event	FR.14. List term
FR.7. Add vocabulary	FR.15. Manage status term
FR.8 Modify vocabulary	FR.16. Send master data message

reference. Table 3 gathers this relationship. It is important to highlight that one of the FCD's requirements can be bound to one or several requirements of the reference set.

Moreover, for each of the identified FCD functional requirements, a description using a template for the rigorous definition of the functional requirements (see Table 2) will be generated. This, when required, will make easier the definition of the corresponding test cases to be used during measurement in activity 4.

As aforementioned, it is necessary to design specific test cases when required for the calculation of some metrics. To describe the test cases, it is highly recommended to use a template, such as the one introduced in Table 4.

In *Activity 2. Specifying the evaluation*, the specific evaluation model is to be selected and customised. In this case study, which is aimed at evaluating the software quality characteristics of 'functional requirement', we followed the recommendations specified in Appendix.

In *Activity 3. Designing the evaluation*, the specifics for the evaluation of the FCDs are addressed. This includes the following information:

- Information about the FCD including the technology (J2EE) and the version of the product (FCD v1.3)
- The purpose of the evaluation: determine whether the FCD is ISO 8000 parts 100–140 or not
- The quality requirements: in this case study, quality requirements are shown in Tables 3 and 4
- The assets to be evaluated: which include all the modules introduced in Fig. 2
- The procedure of the evaluation and a description of the quality model used in the evaluation: the process described in Appendix
- The technological environment used in the evaluation, including all the hardware and software to execute the test cases and to make the calculation to compute the metrics
- An activity plan with estimated and real dates for the evaluation
- The human resources involved in the evaluation
- And finally, the results expected from the evaluation

In *Activity 4. Performing the evaluation*, in the following section, we introduce the calculation of the metrics and the properties (please, refer to Fig. 6 in Appendix for further explanation on the meaning of the acronyms for the properties):

Calculation of the *Functional Implementation Completeness (COM_FUN_IMP)* property: Two metrics are necessary to perform this calculation:

1. Total number of requirements from the requirements set of reference
2. Total of implemented requirements from the requirements set of reference

Table 2 FCD requirement description

<i>FCD ID requirement</i>	<i>FR.7.</i>	<i>Version</i>	<i>1.0</i>	<i>ID in the requirement set of reference</i>	<i>RF.11.</i>
<i>Stakeholders</i>	User			<i>Priority</i>	High
<i>Name</i>	Add Vocabulary				
<i>Description</i>	FCD lets the addition of several vocabularies containing the information about a real-world entity within an organisation. Each vocabulary must have the name, description, organisation, language, and status				
<i>Related FCD requirements</i>	FR.8., FR.9. and FR.10.				

Table 3 Relationship between FCD functional requirements and the requirements set of reference considered for the software product evaluation for MDM-based applications

FCD functional requirements	Requirements set of reference
FR.4. Create master data message <i>Not included in FCD</i> <i>Not included in FCD</i>	FR.1. Build master data message FR.2. Code and/or decode master data message and records FR.3. Verify correctness of master data message FR.4. Record messages of provenance-related events
FR.1. Create provenance event FR.4. Create master data message FR.5. Add header to event FR.6. Add data to event	
FR.4. Create master data message FR.5. Add header to event FR.6. Add data to event	FR.5. Record provenance-related events in the records
FR.2. Create accuracy event FR.4. Create master data message FR.5. Add header to event FR.6. Add data to event	FR.6. Record accuracy-related events in the records
<i>Not included in FCD</i> FR.3. Create completeness event FR.4. Create master data message FR.5. Add header to event FR.6. Add data to event	FR.7. Validate accuracy-related events through an accuracy method FR.8. Record completeness-related events in the records
<i>Not included in FCD</i>	FR.9. Validate completeness-related events through a method of completeness
FR.11. Add term FR.12. Modify term FR.13. Delete term FR.14. List term FR.15. Manage status term	FR.10. Query the data dictionary
FR.7. Add vocabulary FR.8. Modify vocabulary FR.9. Delete vocabulary FR.10. List vocabularies	FR.11. Create a new property in the data dictionary
FR.11. Add term FR.12. Modify term	FR.12. Assign a value to a property
FR.16. Send master data message	FR.13. Exchange master data message

After the study of the documentation, user guide, requirements definition of FCD (see Table 1), and the use of FCD, it was determined that the total number of requirements from the set of reference is 13, so this metric takes a value of 13, and the implemented requirements are 9, so this metric takes a value of 9 (see Table 5). Applying the procedure described in (Rodríguez et al. 2016), and based on the metrics obtained, we concluded that the property *COM_FUN_IMP* takes a value of 69.

Calculation of the *Functional Test Completeness (COM_FUN_TEST)* property: Two metrics are necessary to perform this calculation:

1. Total number of the implemented requirements from the requirements set of reference
2. Total number of the tested requirements from the requirements set of reference

To take a value for these metrics, the support of the test cases documentation (see example in Table 4) is necessary. Using this documentation, checking the test cases in the source code and examining the coverage of the FCD code tested, it was determined that:

Table 4 Example test case for FR.3. Create a new property in the data dictionary

<i>FCD ID requirement</i>	FR.7.	<i>Version</i>	1.0
<i>ID in the requirement set of reference</i>	FR.11.	<i>Test case ID</i>	FR.7.-Test01
<i>Name</i>	Add vocabulary		
<i>Aim</i>	Verify that the properties for a real-world entity in the data dictionary are successfully created.		
<i>Preconditions</i>	None		
<i>Actions</i>	1. Select the option ‘Vocabulary’ in the menu bar 2. Select the subsection ‘Add’ 3. Introduce the name, description, organisation, status and language of the new vocabulary 4. Push ‘Add Vocabulary’ button		
<i>Expected result</i>	FCD shows a confirmation message and it redirects the user to a screen with the list of vocabularies.		
<i>Input data</i>	Name: ‘testName’ Description: ‘This description is an example’ Organisation: ‘3281974’ Language: ‘EN’ Status: ‘ACTIVE’		
<i>Result obtained</i>	–		
<i>Observations</i>	–		
<i>Date</i>	dd/mm/yyyy	<i>Result of test</i>	Success/fail

Table 5 COM_FUN_IMP property: specified and implemented requirements

Reference set requirement ID	FCD requirement ID	Is implemented?
FR.1.	FR.4. Create master data message	✓
FR.2.	<i>Not currently implemented in FCD</i>	×
FR.3.	<i>Not currently implemented in FCD</i>	×
FR.4.	FR.1. Create provenance event	✓
	FR.4. Create master data message	✓
	FR.5. Add header to event	✓
	FR.6. Add data to event	✓
FR.5.	FR.4. Create master data message	✓
	FR.5. Add header to event	✓
	FR.6. Add data to event	✓
FR.6.	FR.2. Create accuracy event	✓
	FR.4. Create master data message	✓
	FR.5. Add header to event	✓
	FR.6. Add data to event	✓
FR.7.	<i>Not currently implemented in FCD</i>	×
FR.8.	FR.3. Create completeness event	✓
	FR.4. Create master data message	✓
	FR.5. Add header to event	✓
	FR.6. Add data to event	✓
FR.9.	<i>Not currently implemented in FCD</i>	×
FR.10.	FR.11. Add term	✓
	FR.12. Modify term	✓
	FR.13. Delete term	✓
	FR.14. List term	✓
	FR.15. Manage status term	✓
FR.11.	FR.7. Add vocabulary	✓
	FR.8. Modify vocabulary	✓
	FR.9. Delete vocabulary	✓
	FR.10. List vocabularies	✓
FR.12.	FR.11. Add term	✓
	FR.12. Modify term	✓
FR.13.	FR.16. Send master data message	✓

Table 6 COM_FUN_TEST property: implemented and tested requirements

Reference set requirement ID	FCD requirement ID	Is implemented?	Is tested?
FR.1.	FR.4. Create master data message	✓	✓
FR.2.	<i>Not currently implemented in FCD</i>	×	×
FR.3.	<i>Not currently implemented in FCD</i>	×	×
FR.4.	FR.1. Create provenance event	✓	✓
	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	×
	FR.6. Add data to event	✓	✓
FR.5.	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	✓
	FR.6. Add data to event	✓	✓
FR.6.	FR.2. Create accuracy event	✓	✓
	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	×
	FR.6. Add data to event	✓	✓
FR.7.	<i>Not currently implemented in FCD</i>	×	×
FR.8.	FR.3. Create completeness event	✓	✓
	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	×
	FR.6. Add data to event	✓	✓
FR.9.	<i>Not currently implemented in FCD</i>	×	×
FR.10.	FR.11. Add term	✓	✓
	FR.12. Modify term	✓	✓
	FR.13. Delete term	✓	✓
	FR.14. List term	✓	×
	FR.15. Manage status term	✓	×
FR.11.	FR.7. Add vocabulary	✓	✓
	FR.8. Modify vocabulary	✓	✓
	FR.9. Delete vocabulary	✓	✓
	FR.10. List vocabularies	✓	✓
FR.12.	FR.11. Add term	✓	✓
	FR.12. Modify term	✓	✓
FR.13.	FR.16. Send master data message	✓	✓

1. Not all the functionalities specified in the requirements set of reference were implemented
2. Not all the functionalities from the requirements set of reference were tested

Consequently, it was determined that the total number of requirements from the set of reference is 9, so this metric takes a value of 9, and the total number of tested requirements from the requirements set of reference is 5, therefore, this metric takes a value of 5. Applying the procedure described in (Rodríguez et al. 2016), and based on the metrics obtained, we reached the conclusion that the property *COM_FUN_TEST* takes a value of 38 (Table 6).

Calculation of the *Functional Implementation Completeness (CORR_FUN_IMP)* property: Two metrics are necessary to perform this calculation:

1. Total number of successfully tested requirements from the requirements set of reference
2. Total of implemented requirements from the requirements set of reference

To take a value for these metrics the support of the test cases description (see example in Table 7) is essential.

Table 7 FR.7 Test case execution summary (the differences from the one shown in Table 5 are the rows entitled ‘Result obtained’, ‘Observations’, ‘Date’ and ‘Result of test’)

<i>ID</i>	FR.7.	<i>Version</i>	1.0
<i>ID in the requirements set of reference</i>	FR.11.	<i>Priority</i>	High
<i>Name</i>	Add vocabulary		
<i>Aim</i>	Verify that the properties for a real-world entity in the data dictionary are successfully created.		
<i>Preconditions</i>	None		
<i>Actions</i>	1. Select the option ‘Vocabulary’ in the menu bar 2. Select the subsection ‘Add’ 3. Introduce the name, description, organisation, status and language of the new vocabulary 4. Push ‘Add Vocabulary’ button		
<i>Expected results</i>	FCD shows a confirmation message and it redirects the user to a screen with the list of vocabularies.		
<i>Input data</i>	Name: ‘testName’ Description: ‘This description is an example’ Organisation: ‘3281974’ Language: ‘EN’ Status: ‘ACTIVE’		
<i>Result obtained</i>	FCD redirects automatically to the webpage that contains all the vocabularies and shows the vocabulary recently added.		
<i>Observations</i>	None		
<i>Date</i>	10/09/2018	<i>Result of test</i>	Success

Table 8 CORR_FUN_IMP property: specified, implemented and tested successfully requirements

Reference set requirement ID	FCD requirement ID	Is it implemented?	Is it successfully tested?
FR.1.	FR.4. Create master data message	✓	✓
FR.2.	<i>Not currently implemented in FCD</i>	×	×
FR.3.	<i>Not currently implemented in FCD</i>	×	×
FR.4.	FR.1. Create provenance event	✓	✓
	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	×
	FR.6. Add data to event	✓	✓
FR.5.	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	✓
	FR.6. Add data to event	✓	✓
FR.6.	FR.2. Create accuracy event	✓	✓
	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	×
	FR.6. Add data to event	✓	✓
FR.7.	<i>Not currently implemented in FCD</i>	×	×
FR.8.	FR.3. Create completeness event	✓	✓
	FR.4. Create master data message	✓	✓
	FR.5. Add header to event	✓	×
	FR.6. Add data to event	✓	✓
FR.9.	<i>Not currently implemented in FCD</i>	×	×
FR.10.	FR.11. Add term	✓	✓
	FR.12. Modify term	✓	✓
	FR.13. Delete term	✓	✓
	FR.14. List term	✓	×
	FR.15. Manage status term	✓	×
FR.11.	FR.7. Add vocabulary	✓	✓
	FR.8. Modify vocabulary	✓	✓
	FR.9. Delete vocabulary	✓	×
	FR.10. List vocabularies	✓	✓
FR.12.	FR.11. Add term	✓	✓
	FR.12. Modify term	✓	×
FR.13.	FR.16. Send master data message	✓	✓

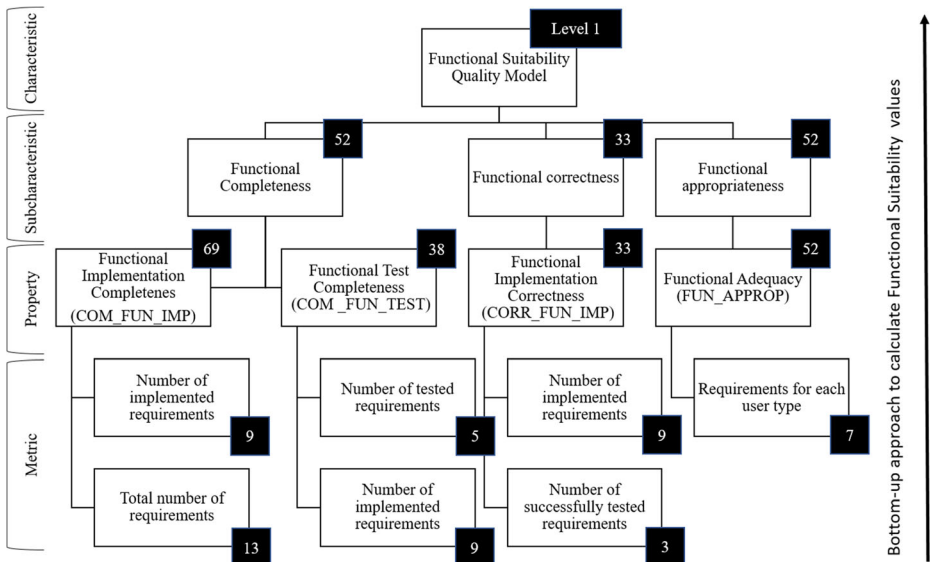


Fig. 3 Summary of the bottom-up procedure and values for each of the metrics, properties, subcharacteristics and functional suitability in the first evaluation

Using this documentation, examining the test cases in the source code and checking the coverage of the code tested of FCD, it was determined that not all the functionalities from the requirements set of reference were tested successfully. Hence, it was determined that the total number of successfully tested requirements from the requirements set of reference is 3, so this metric takes a value of 3, and the implemented requirements are 9, so this metric takes a value of 9 (see Table 8). Applying the procedure described in (Rodríguez et al. 2016), and based on the metrics obtained, we concluded that the property *COM_FUN_IMP* takes a value of 33.

Calculation of the *Functional Adequacy (FUN_APPROP)* property: it is necessary to determine the number of requirements not perceived as appropriate for use in the scope of the evaluation. To put it another way, the number of requirements that were identified in FCD, but since they are not relevant for use, they have not been described in the functional requirements specification. For this case, after the study of the documentation of FCD, and considering that only 16 functional requirements out of the 60 (the whole of functional requirements in FCD) were mapped with the reference set of requirements, it was determined

Table 9 Relationship between new FCD functional requirements and the requirements set of reference

FCD functional requirements	Requirements set of reference
FR.17. Code Master Data Messages	FR.2. Code and/or decode master data message and records
FR.18 Decode Master Data Messages	
FR.19. Manage correctness methods	FR.3. Verify correctness of master data message
FR.20. Validate correctness-related events	
FR.21. Manage accuracy methods	FR.7. Validate accuracy-related events through an accuracy method
FR.22. Validate accuracy-related events	
FR.23. Manage completeness methods	FR.9. Validate completeness-related events through a method of completeness
FR.24. Validate completeness-related events	

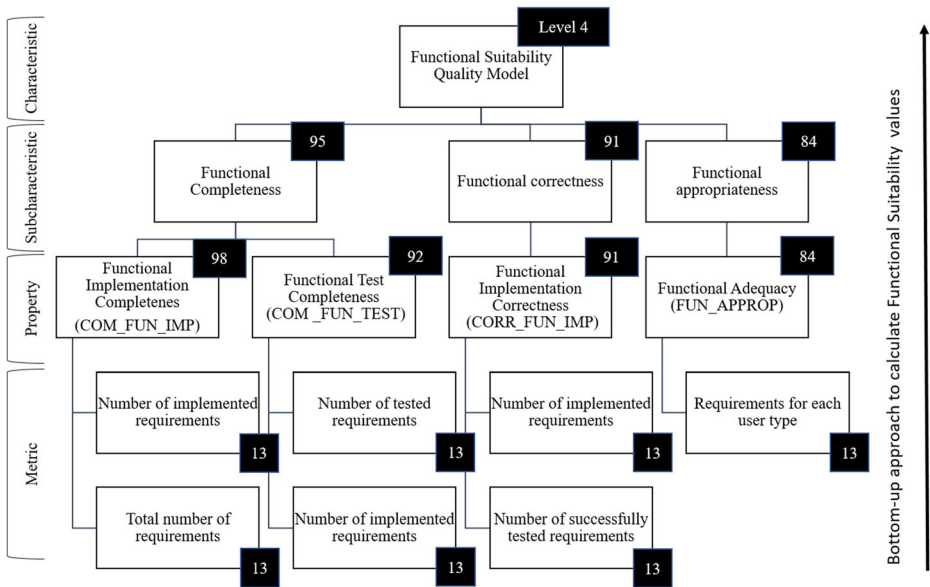


Fig. 4 Summary of the bottom-up procedure and values for each of the metrics, properties, subcharacteristics and functional suitability in the second evaluation

that this metric takes a value of 7. Applying the procedure described in (Rodríguez et al. 2016), and based on the metrics obtained, we reached the conclusion that the property *FUN_APPROP* takes a value of 52.

Throughout the application of the bottom-up approach (see Appendix), the value for each subcharacteristic was calculated, and after that, it was determined the functional suitability level applying a profile function, as described in (Rodríguez et al. 2016). In Fig. 3, all the values for each one of the elements from the quality model and the level of functional suitability for FCD are specified. In this case, and after computing the metrics through the quality model, it was determined that the level of functional suitability for FCD is 1: the lowest level.

Finally, during *Activity 5. Finishing the evaluation*, the evaluation report was generated including the results obtained as the result in activity 4. This report

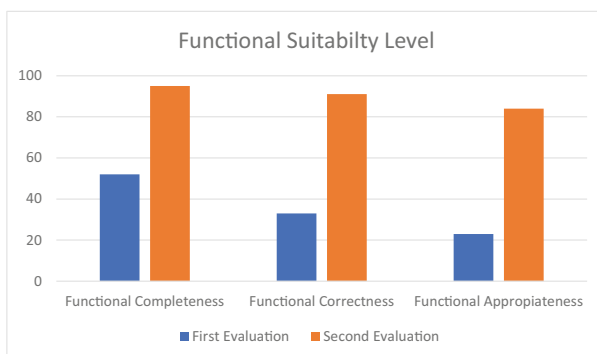


Fig. 5 Results of the first and the second evaluations

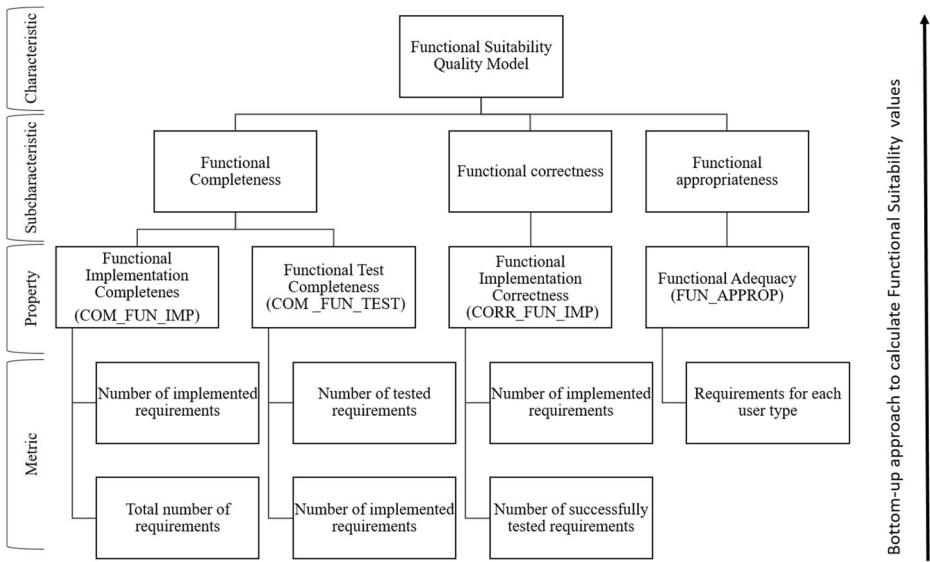


Fig. 6 Elements that compose the functional suitability quality model (extracted and adapted from (Rodríguez et al. 2016))

indicates the quality level obtained, in this case 1. The rationale for these results is based on the fact that not all the requirements of the reference set of requirements are implemented in FCD, and not all the result for each of the quality subcharacteristics were enough for achieving a higher level. In addition, an improvement report that includes a plan of actions to amend the functional adequacy of FCD was created. Some of these improvement actions are:

1. *Implementation of the functional requirements from the set of reference not implemented in FCD:* FR.2. Code and/or decode master data message and records, FR.3. Verify correctness of master data message, FR.7. Validate accuracy-related events through an accuracy method and FR.9. Validate completeness-related events through a method of completeness.
2. *Definition of test cases for all the functionalities of the set of reference not defined.*

After the generation of the report, all the changes were conducted by OX. Firstly, new functional requirements in FCD were identified to support the functional requirement from

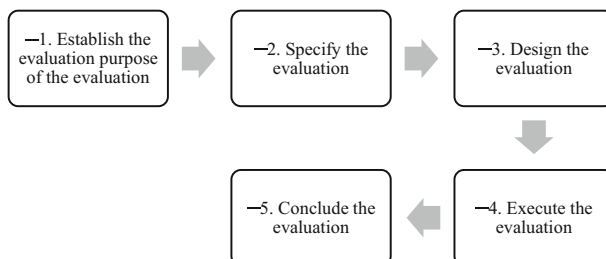


Fig. 7 Activities of the evaluation process for functional suitability

the set of functional requirements inferred from ISO 8000 parts 100 to 140 (see Table 9) that were not previously implemented.

Secondly, all these functionalities were implemented. Thirdly, all the test cases were reviewed for each functionality, and the test cases for the functionalities that do not previously exist, they were designed, implemented and documented accordingly. Finally, all the test cases—for all functionalities—were executed, and it was examined that the results obtained were the findings specified and expected in the test cases definition.

After making all the suggested improvements on FCD, a new version of FCD was generated. This version of the product will be evaluated in the second evaluation presented in Sect. 5.1.3.

5.3 Second evaluation of MDM-based application

For the second evaluation, the evaluation procedure did not include the two first activities, as they did not change since the first evaluation. Therefore, this second evaluation began with the activity 3. The following tasks were performed.

During *Activity 3. Design the Evaluation*, the action plan to conduct the second evaluation was generated. This second plan was the same as the plan developed in activity 3 from the first evaluation. As a result of the experience of the first evaluation, the times indicated for the second evaluation in this plan are shorter than the times indicated in the plan of the first evaluation.

During *Activity 4. Performing the evaluation*, the new version of FCD was measured, following tasks planned in activity 3 and performing similar tasks that were conducted in activity 4 of the first evaluation. After the improvement's execution, the values of the metrics provided during this evaluation were higher because:

1. All the functionalities specified in the reference set of requirements were implemented, and therefore, the value of the metric focused on the *number of implemented requirements* is better (higher) than that obtained in the first evaluation
2. All the functionalities specified in the reference set of requirements had assigned a test case, so the value of the metric focused on the *number of tested requirements* is better (higher) than that obtained in the first evaluation
3. All the test cases were executed, and it was examined that the results obtained were the specified and expected in the test cases definition, and hence, the value of the metric focused on the *number of implemented requirements* is better (higher) than that obtained in the first evaluation
4. The number of FCD functional requirements is higher, and there are more functional requirements mapped with the reference set of requirements, so the value of the metric focused on the *number of requirements for each user type* is better (higher) than that obtained in the first evaluation

After computing the metrics, properties and subcharacteristics, the quality level for functional suitability took a higher value of 4. The results obtained from measuring FCD in the second evaluation are shown in Fig. 4.

Finally, during *Activity 5. Finishing the evaluation*, a new evaluation report was produced. After the amendments conducted by OX, FCD reached level 4. The fact of achieving this level implies that the quality level of FCD is considerably good, but not perfect. However, all the functionalities from the requirements set of reference were in one these states 'implemented', 'tested' and 'successfully tested'; therefore, it is not strictly necessary to create a new

improvement report. As it can be seen from the results of the values of metrics obtained during the second evaluation (see Fig. 4), OX—in the new version of FCD—solved the problems identified in the first evaluation. This improved the results of the metrics and consequently, the results of the evaluation of functional suitability (see Fig. 5).

6 Conclusions

Analysing the most important points that we have found when running the case study; we reached the following conclusions:

- The lack of updated documentation of the FCD development. This fact was determinant not only to identify the requirements that must be included in the evaluation but also to design the test cases necessary to run the evaluation.
- Regarding the set of reference of functional requirements: we discovered that the achieved efforts to infer the requirements from ISO 8000 covered all parts except ISO 8000-115, ISO 8000-116 and ISO 8000-150. In future works, we will extend the current version of the reference set, with some other requirements inferred from ISO 8000-115 and ISO 8000-116. The reason why we have not included ISO 8000-115 and ISO 8000-116 in this part is based on the fact that these two parts are aimed at defining how the identifiers of some types of master data should be built, and we did not conceive it as if they were functional requirements, but rather non-functional. On the other hand, ISO 8000-150 deals with some concerns related to master data management, which is out of the scope of our proposal.
- It is of prime importance to recall that in the case of the FCD, it may happen that one or several software functional requirements were mapped to only one functional requirement in the set of reference. When it concerns proposed improvements to the evaluated systems, it will be necessary to maintain traceability of the original requirement(s) to the one(s) in the set of reference, to better identify which original requirement should be improved.
- The improvements achieved to the FCD have undoubtedly benefitted not only the system per se but also some organisational concerns. For example, the security in the exchange of master data messages between the master data-based applications and the master data servers, the verification of the message in relation to the events are now conducted, what enables the traceability of the master data through the whole data lifecycle, just to name a few.

After analysing the experience acquired during the case study, we can conclude that evaluating the functional suitability as presented in ISO 25010 is a truly useful approach with respect to improving the performance of the master data-based applications.

Funding information This research is partially funded by Industrial PhD DIN2018-009705, funded by the Spanish Ministry of Science, Innovation and Universities, GEMA: Generation and Evaluation of Models for dAta Quality (Ref.: SBPLY/17/180501/000293), DQIoT project (INNO-20171086 EUREKA Project No. E!11737), funded by CDTI, ECD project (PTQ-16-08504), funded by the ‘Torres Quevedo’ Program of the Spanish Ministry of Economy, Industry and Competitiveness, TESTIMO project (Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla-La Mancha, and Fondo Europeo de Desarrollo Regional FEDER, SBPLY/17/180501/000503), and ECLIPSE project (RTI2018-094283-B-C31) funded by Ministry of Science, Innovation and Universities and FEDER funds.

Appendix

A certification environment for software quality based on ISO 25010 and ISO 25040

The environment for evaluation and certification of the functional suitability in software product quality against ISO 25010 is presented by Rodriguez et al. in (2015, 2016). This environment is used to evaluate and certify that a software product meets the functional requirements, and therefore, fulfils the purpose for which it was created. According to <https://www.iso25000.com/index.php/en/certified-products>, the environment has been used to certify more than 20 software products in different business areas: health, human resources, education, business intelligence, or risk management. However, this environment has not yet been used to certify MDM-based applications because of the specifics of this type of system, even an increasing demand of this service (Forrester 2019; Gartner 2018). This environment consists of a software quality model (which includes ‘functional suitability’ as introduced in ISO 25010 (ISO 2011a)), and an evaluation process based on ISO 25040 (ISO 2011b).

Functional suitability quality model

The software quality model contains the set of characteristics and subcharacteristics of the quality against which to evaluate a software product. As aforementioned, one of these software quality characteristics is ‘functional suitability’, which represents the ability of the software product to provide functions that meet the needs (stated and implied), when the product is used in specified conditions. This characteristic is split into the three following subcharacteristics:

- ‘Functional completeness’ is the degree to which the set of functions covers all the specified tasks and user objectives.
- ‘Functional correctness’ is the degree to which a product or system provides the correct results with the needed degree of precision.
- ‘Functional appropriateness’ is the degree to which the functions facilitate the accomplishment of specified tasks and objectives.

In addition, each one of the characteristics is split into one or more properties (see Fig. 6) that are used to evaluate the characteristic, and each property uses several metrics in order to calculate the value of the property. These metrics were extracted from the systematic review purposed by Blanco et al. in (Blanco et al. 2012).

Functional suitability evaluation process

The evaluation process for software products certification needs the evaluation of the software quality characteristics. For the sake of the replicability and accuracy of the results, the evaluation process specified in ISO 25040 (ISO 2011b) is encouraged. The evaluation includes the five activities represented in Fig. 7.

The main goal of the first activity is to establish the requirements and scope for the evaluation. During this activity, there are several meetings with stakeholders to present the

evaluation process, the evaluation needs, and to determine the main characteristics and documentation about the MDM-based application aim of the evaluation. Additionally, in this first activity, the set of functional requirements to be met by an MDM-based application compliance to ISO 8000 parts 100 to 140 is specified, and the functional requirements of the MDM-based application with this set of reference is mapped. In the second activity, the main goal is to specify the evaluation. The third activity is aimed at defining the goal and planning for the evaluation. The plan should consider available resources for the evaluation. In the fourth activity, the main goal is to execute the evaluation activities according to the evaluation plan. Finally, the fifth and last activities consist of issuing the report with the results of the evaluation. This result of the evaluation should be informed to the applicant of the evaluation and those interested in this final activity.

Given the importance of activity 4, it is worth to further describing its goal. The evaluation process is performed by following a bottom-up approach, which begins by calculating the metrics identified at the bottom of Fig. 6 (number of requirements, number of requirements implemented, number of requirements tested, and requirements for user type). Some of these values can be calculated based on the execution of customised testing cases. The possible values of all these metrics are normalised and they range [0,100]. The values of these metrics are used to compute the properties (e.g. functional implementation completeness) defined in the immediately higher level. The possible values of these properties are also normalised and range [0, 100]. Analogously, the value of the properties is used to calculate the value of the subcharacteristics (e.g. functional completeness). The value of the subcharacteristics is also normalised and range [0,100]. Finally, after calculating the subcharacteristic values, it is necessary to compute these results to obtain a value for the functional suitability. The procedure to compute the metrics to determine the quality level of each quality characteristic and the quality level for the evaluation of the functional adequacy of a software product are available in (Rodríguez et al. 2016). The quality level of functional suitability is represented on a level scale expressed in a range from 1 to 5, where 1 is the lowest level and 5 is the highest.

References

- Allen, M., & Cervo, D. (2015). *Multi-domain master data management: Advanced MDM and data governance in practice*. Morgan Kaufmann.
- Blanco, A., Reales, P. & Rodríguez, M. (2012). Metrics to evaluate functional quality: A systematic review., *Proceedings of Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI 2012)*, Madrid, Spain.
- Choi, M.-Y., Moon, C.-J., Park, K.-S. & Baik, D.-K. (2010) An enterprise master data model based on the data taxonomy based on their origin, *Proceeding of International Conference on Enterprise Information Systems and Web Technologies (EISWT 2010)*, pp. 34–41.
- Cleven, A. & Wortmann, F. (2010). Uncovering four strategies to approach master data management, *Proceeding of the 43rd Hawaii International Conference on System Sciences 2010*, pp. 1–10.
- Dreibelbis, A., Hechler, E., Milman, I., Oberhofer, M., van Run, P., & Wolfson, D. (2008). *Enterprise Master Data Management (Paperback): An SOA Approach to Managing Core Information*. Pearson Education.
- Fleckenstein, M., & Fellows, L. (2018). *Modern Data Strategy*. Springer International Publishing.
- Forrester (2019) The Forrester Wave: Master Data Management, Q1 2019. <https://cutt.ly/1ricvfb> Last accessed in December 2019.
- Gartner (2018) Magic Quadrant for Master Data Management Solutions. <https://cutt.ly/Fricny3> Last accessed in December 2019.
- ISO (2009), *ISO 8000-110:2009 Data quality – Part 110: Master data: Exchange of characteristic data: Syntax, semantic encoding, and conformance to data specification*. International Standardization for Organization (ISO).

- ISO (2011a), *ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. ISO/IEC.
- ISO (2011b), *ISO/IEC 25040:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Evaluation process*. ISO/IEC.
- ISO (2011c), *ISO 8000-150:2011 Data quality – Part 150: Master data: Quality management framework*. International Standardization for Organization.
- ISO (2016a), *ISO 8000-100:2016 Data quality – Part 100: Master data: Exchange of characteristic data: Overview*. International Standardization for Organization.
- ISO (2016b), *ISO 8000-120:2016 Data quality – Part 120: Master data: Exchange of characteristic data: Provenance*. International Standardization for Organization.
- ISO (2016c), *ISO 8000-130:2016 Data quality – Part 130: Master data: Exchange of characteristic data: Accuracy*. International Standardization for Organization.
- ISO (2016d), *ISO 8000-140:2016 Data quality – Part 140: Master data: Exchange of characteristic data: Completeness*. International Standardization for Organization.
- ISO (2018a), *ISO 8000-115:2018 Data quality – Part 115: Master data: Exchange of quality identifiers: Syntactic, semantic and resolution requirements*. International Standardization for Organization.
- ISO (2019), *ISO 8000-116 Data quality – Part 116: Application of ISO 8000-115 to the formatting of Authoritative Legal Entity Identifiers (ALEI) for individuals and organizations*. International Standardization for Organization.
- Loshin, D. (2010). *Master Data Management*. Morgan Kaufmann.
- Mahanti, R. (2019). *Data quality: Dimensions, Measurement, Strategy, Management, and Governance*. ASQ Quality Press.
- Otto, B., Ebner, V. & Hüner, K. M. (2010) Measuring master data quality: Findings from a case study. *Proceedings of 16th Americas Conference on Information Systems (AMCIS 2010)*, 5, 3761–3769.
- Otto, B., Hüner, K. M., & Österle, H. (2012). Toward a functional reference model for master data quality management. *Information Systems and e-Business Management*, 10(3), 395–425.
- Rivas, B., Merino, J., Caballero, I., Serrano, M., & Piattini, M. (2017). Towards a service architecture for master data exchange based on ISO 8000 with support to process large datasets. *Computer Standards & Interfaces*, 54(2), 94–104.
- Rodríguez, M., Piattini, M., & Fernandez, C. M. (2015). A hard look at software quality: Pilot program uses ISO/IEC 25000 family to evaluate, improve and certify software products. *Quality Progress*, 48(9), 30–36.
- Rodríguez, M., Oviedo, J. R., & Piattini, M. (2016). Evaluation of software product functional suitability: A case study. *Software Quality Professional Magazine*, 18(3), 18–29.
- Smith, H. A., & McKeen, J. D. (2008). Developments in practice XXX: master data management: salvation or snake oil?. *Communications of the Association for Information Systems*, 23(1), 63–72.
- Spruit, M., & Pietzka, K. (2015). MD3M: The master data management maturity model. *Computers in Human Behavior*, 51, 1068–1076.
- Talbur, J. R. (2011). *Entity resolution and information quality*. Elsevier.
- Zúñiga, D. V., Cruz, R. K., Ibañez, C. R., Domínguez, F. & Moguerza, J. M. (2018) Master data management maturity model for the microfinance sector in Peru. *ACM International Conference Proceeding Series*, 49–53.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Fernando Gualo received the B.Sc and M.Sc. degree in computer science from the University of Castilla-La Mancha. He is currently founder and CEO of the spin-off of the University of Castilla-La Mancha DQTeam. Certified Information System Auditor (CISA) by the Information System Audit and Control Association (ISACA) since 2017, and Software Engineering Auditor for AENOR since 2018. He is part of the ALARCOS Research Group since 2015. His research interests include master data management and data quality.



Ismael Caballero received his M.Sc and Ph.D Degree in Computer Science in University of Castilla-La Mancha (Spain) in 2004. He works as Associate Professor in the Information Systems and Technologies Department of UCLM from 2001, and he was appointed as Training Head of the spinoff DQTeam in 2017. He has been researching on data quality management and data governance since 1998, co-authoring several conference and journal papers. He teaches data quality management and data governance foundations in many universities and companies. Caballero holds CISA certification by ISACA from 2016 and CDO-1 certification by UALR-MIT since 2017. He is currently a member of ISO TC184/SC4 and a project editor of several parts of ISO 8000-60 series development project as well as project editor of ISO 8000-62



Moisés Rodríguez is a M.Sc. and has a PhD in Computer Science from the University of Castilla- La Mancha. He is founder and CEO of the international accredited laboratory for software quality assessment AQCLab, is a Certified Information System Auditor for the Information System Audit and Control Association (ISACA), and a Lead Software Engineering and R&D Auditor for AENOR. He is also a part-time Professor with the Escuela Superior de Informática of the University of Castilla-La Mancha. He is part of the ALARCOS Research Group. His research interests include software process, software product and data quality assessment, improvement and certification.