Validating Data Quality Actions in Scoring Processes

C. CAPPIELLO, C. CERLETTI, C. FRATTO, and B. PERNICI, Politecnico di Milano

Data quality has gained momentum among organizations upon the realization that poor data quality might cause failures and/or inefficiencies, thus compromising business processes and application results. However, enterprises often adopt data quality assessment and improvement methods based on practical and empirical approaches without conducting a rigorous analysis of the data quality issues and outcome of the enacted data quality improvement practices. In particular, data quality management, especially the identification of the data quality dimensions to be monitored and improved, is performed by knowledge workers with potentially limited skills and experience. Control methods are therefore designed on the basis of expected and evident quality problems; thus, these methods may not be effective in dealing with unknown and/or unexpected problems. This article aims to provide a methodology, based on fault injection, for validating the data quality actions used by organizations. We show how it is possible to check whether the adopted techniques properly monitor the real issues that may damage business processes. At this stage, we focus on scoring processes, i.e., those in which the output represents the evaluation or ranking of a specific object. We show the effectiveness of our proposal by means of a case study in the financial risk management area.

CCS Concepts: • **Information systems** → **Uncertainty**; **Data analytics**; • **Applied computing** → *Business process management*; *Enterprise data management*;

Additional Key Words and Phrases: Data quality, assessment, decision support, decision processes

ACM Reference format:

C. Cappiello, C. Cerletti, C. Fratto, and B. Pernici. 2017. Validating Data Quality Actions in Scoring Processes. *J. Data and Information Quality* 9, 2, Article 11 (December 2017), 27 pages.
https://doi.org/10.1145/3141248

1 INTRODUCTION

Issues related to Data Quality (DQ) often have negative repercussions on business processes. In fact, their effects may result in economic and efficiency losses at both operational and strategic levels. According to the Garbage In Garbage Out (GIGO) concept, poor-quality output data can be caused by poor-quality input data but also by poor-quality data processing. Therefore, in order to address poor-quality data issues, controls on the input data and in-process data are needed. The former controls aim to ensure that input data are adequate for starting the process activities. The latter have the goal of assessing whether the data manipulated and generated by the process are characterized by a sufficient quality level.

In general, several dimensions are considered for quality evaluation and have been widely studied in the literature; most of the proposed assessment algorithms are characterized by a subjective nature (Batini and Scannapieco 2016; Kritikos et al. 2013). Data Quality Controls (DQC)

Authors' addresses: Politecnico di Milano, piazza Leonardo da Vinci, 32, 20133 Milano, Italy; email: cinzia.cappiello@polimi.it.

https://doi.org/10.1145/3141248

can be performed by adding monitoring activities to the process to evaluate the traditional DQ dimensions and measures (e.g., Pipino et al. (2002) and Wang and Strong (1996)) and to take action if the quality level is not acceptable for the process. The design of quality controls for assessing and monitoring DQ levels is up to knowledge workers who, on the basis of their background and experience, identify the DQ dimensions to consider. These evaluations are often biased, however, since some details on processing activities may be unknown to the knowledge workers, in particular in large and complex processes. Thus, they could overestimate or underestimate the quality values and relative importance of data quality dimensions on the process outcome. Currently, to the best of our knowledge, there is no systematic methodology for the validation of DQ assessment techniques in the literature. As a result, objective criteria are needed to establish whether such assessment techniques properly monitor real issues that could affect business processes.

The goal of this article is to propose a methodologyn *Data Quality Validation Methodology (DQVM)*, as a systematic approach based on objective and experimental results to evaluate the consequences of poor data quality and related DQ actions on the outcome of the processes. DQVM proposes a set of steps to analyze the effects of faults, introducing them systematically in the process using fault injection. The goal is to systematically assess the impact of data quality faults in the process outcome: we validate the relevance degree assigned to monitored DQ dimensions, the effectiveness of DQ actions, and the impact of poor data quality items on the process behavior.

We tested the methodology on a specific type of processes with outcome, named *scoring processes*, in which the goal is to provide an assessment (e.g., a ranking or a rating) of a given object based on a given number of inputs and information sources and given evaluation rules. This type of process is common in many business areas—in particular, in the financial domain—in which the institutions are particularly keen on controlling the quality of used data and evaluating the quality of the obtained results.

The article is organized as follows. Section 2 describes DQ issues in business processes and related work. Section 3 describes the characteristics and requirements for data quality in scoring processes and introduces a running example that we use to explain the different methodological steps. Sections 4 and 5 present the different phases of the DQVM methodology. The application of the methodology to a case study in the financial risk management sector is presented in Section 6. Our conclusions are presented in Section 7.

2 RELATED WORK

Data quality is becoming increasingly more important within business contexts. This is due to the fact that the amount of information that organizations have to manage is steadily increasing. However, such data are often not characterized by an appropriate level of quality. In general, problems related to DQ can originate in different causes (Strong et al. 1997a), such as multiple and contradicting sources, loss of data caused by systemic errors in the data-generation process, difficulty in accessing information in a reasonable time due to volumes of stored data that are too large, and workarounds (Schmidt et al. 2016), e.g., postfactum information changes and fictitious entity instances in databases. In order to address such issues, it is necessary to evaluate the quality of available information and define appropriate improvement actions. This is very important, as poor data quality in business environments can generate very high costs and inefficiencies (Redman 1998). For this reason, some organizations are establishing specific working units and internal services to manage data quality most efficiently.

In general, data quality assessment and improvement activities are included in the so called *Data Quality Management*, which is divided into four phases: (i) definition of DQ dimensions to evaluate; (ii) measurement of defined DQ dimensions; (iii) analysis of poor-quality root causes; and (iv) definition of improvement actions. As shown in Batini and Scannapieco (2016), in the literature there are different methodologies able to support data-quality management. Some of these methodologies, referred to as *audit methodologies* (e.g., AIMQ (Lee et al. 2002), DQA (Pipino et al. 2002) and QAFD (De Amicis and Batini 2004)), are focused on the assessment steps and do not consider the improvement part. Moreover, they mostly base the data-quality level assessment on subjective evaluations and sometimes do not treat DQ as an intrinsic concept, stating instead that quality of data cannot be assessed independently from data consumers (Strong et al. 1997b). In addition, many different DQ dimensions are defined in the literature (e.g., Wang and Strong (1996) and Redman (1996)). In Batini and Scannapieco (2016), an extensive overview of proposed dimensions, their assessment techniques, and DQ improvement methodologies is given. DQ measurement is described in the ISO/IEC 25014:2015 report (BSI 2015) as well. However, also in this case, many measures are declared to be dependent on subjective measurements. As a consequence, techniques for proving their effectiveness are needed.

It is necessary to highlight that most of the literature contributions have a database perspective; only a few papers considered the evaluation of the quality—and, thus, reliability—of the data used and exchanged in a business process. One of the first contributions in this direction is the well-known model Information Product MAPping (IP-MAP), i.e., a modeling language that provides the possibility of highlighting data-quality issues in business processes starting from the assumption that information can be treated as manufacturing products (Shankaranarayanan et al. 2000; Scannapieco et al. 2002). One of the important aspects of IP-MAP is the definition of the data-quality blocks to indicate the points in the process at which data-quality improvement actions have to be taken. Ofner et al. (2012) an in-depth discussion of data quality in business processes and how to integrate a data-quality perspective in processes, about where data should be assessed and which evaluation rules should be used. A thorough discussion of process modeling languages toward process-driven data-quality management is presented in Glowalla and Sunyaev (2014). They consider the insertion of data-quality aspects in process modeling languages as a promising direction.

Other contributions attempt to identify the relevant quality dimensions to consider in business processes. For example, Heravizadeh et al. (2009) relate the quality of business processes with DQ dimensions able to assess the quality of functions, input and output objects, and nonhuman and human resources. Falge et al. (2012) identify a set of DQ dimensions relevant for evaluating data quality in collaborative business processes in business networks. Soffer (2010) highlights the importance of data accuracy, showing potential consequences of data inaccuracies and a way to design robust processes. Considering such contributions, it is possible to state that the most considered data-quality dimensions for business processes are accuracy, completeness, and coherence (i.e., consistency), and timeliness (Batini and Scannapieco 2016; Hazen et al. 2014; Panahy et al. 2014). We focus in this article on all three of these dimensions.

We adopted the following definitions for the chosen dimensions. *Accuracy* is commonly defined as “the extent to which data are correct, reliable and certified” (Wang and Strong 1996). Operationally, in our work, we adopted the definition by which accuracy is a measure of the proximity of a data value *v* to some other value *v’* that is considered correct (Redman 1996). *Completeness* is defined as the degree to which values are included in a data collection (Redman 1996). *Coherence* (Redman 1996), also referred to as *Consistency*, is defined as the degree of satisfaction of integrity constraints and rules defined for the data.

In this article, we validate our work analyzing a large case study from a financial institution, focusing on evaluating the probability of default, which is one of the main components of risk analysis. In the literature, the challenges of data quality in credit risk evaluation processes have been discussed in Moges et al. (2012). In particular, the paper illustrates the main quality dimensions considered relevant by stakeholders and identifies accuracy as the main issue in this context.

The approach proposed in this article is suited not only for evaluation processes in financial domains but is applicable also in other domains in which decisions are taken on the basis of several data sources and an evaluation process. Several recent papers advocate the need of further investigation of data quality in process-based systems. In Hazen et al. (2014), the research issues in data quality in supply chain management, in which processes are defined across multiple organizations, are investigated. In particular, the authors stress the importance of acknowledging, measuring, and monitoring the quality of data and of establishing monitoring schemes to examine the effects of controlling data, as the results of decisions based on poor-quality data could be costly. In the health domain, Berner et al. (2005) systematically analyze how missing data elements resulted in inappropriate and unsafe recommendations in almost 77% of the studied cases. The results show that important gaps in the medical record can affect the accuracy of a decision support system designed to improve safe prescribing. In Berner et al. (2005) and Hausvik (2017) based on a comprehensive literature review, the need of defining organizational performance and process performance with respect to data quality is discussed. Panahy et al. (2014) discuss the need for experimental research to assess impact of quality dimensions on business processes. Evron et al. (2017) consider design-time analysis of potential data inaccuracy based on synchronization points based on dependencies among data.

In order to understand and systematically evaluate the impact of data-quality errors and data quality control actions on the whole business process, we based our work on *fault injection techniques*. Considering that a fault can be defined as a physical defect, imperfection, or trouble that can occur in any hardware or software component (Ziade et al. 2004), fault injection refers to a set of techniques able to verify and test the reliability of a system by introducing some anomalies. In this way, it is possible to understand fault effects and hence the different behaviors that such anomalies can originate without analyzing the structure of the process, as done, for instance, in Meda et al. (2010).

In the literature, fault injection methodologies are applied in different domains, such as hardware and software testing (Hsueh et al. 1997; Ziade et al. 2004) and composed web services (Fugini et al. 2009). In assessing the quality of web services, Fugini et al. (2009) use faults injection to evaluate the impact of the quality of composed services considering data faults and time delays. Data faults can be originated by value mismatches, different formats, missing data, or delays in update operations. Regarding time delays, their impact depends on the fault position within the process structure of web service composition. In this case, faults injection aims to trigger web service timeouts, i.e., mechanisms that are activated if a web service does not respond within predetermined time intervals. Effects on the outcome of the process may vary; Fugini et al. (2009) propose measuring the severity of a fault considering wheter it causes a failure in the service and, in case of data faults, to measure the distance between the origin of the fault and the point of the composed service where it is detected. Fugini et al. (2009) clearly show that system behavior simulation with fault injections is very useful because it makes it possible to detect the system weaknesses and then select the appropriate corrective actions to improve the overall system quality. However, in Fugini et al. (2009), the focus was on assessing the impact of poor data quality on the process execution rather than on assessing its impact on the quality of the result, which is the focus of this article. The advantage of fault injection techniques is that it is possible to analyze the impact on the process of data-quality problems basing the analysis only on the types of faults and on the process structure. Therefore, it is not necessary to model data dependencies, and multiple effects of faults on different data-quality dimensions are also captured.

For this reason, in this article, we adopt a fault-injection approach to provide a systematic assessment of data quality in business processes. Compared to Fugini et al. (2009), the focus of this article is twofold: assessment of the data quality of the outcome of a process and validation of the effectiveness of data-quality assessment and improvement activities when they are inserted into the process to make it more robust in the case of data quality faults. In addition, we aim to identify the DQ dimensions that need to be considered since issues related to such dimensions have a higher negative impact on the process. The innovative aspect of our approach is that the importance of data-quality dimensions is estimated by using simulations, thus by an objective method and not by adopting a subjective approach (e.g., AHP (Saaty 2001) or a user-driven ranking (Debattista et al. 2016)).

3 SCENARIO: SCORING PROCESSES

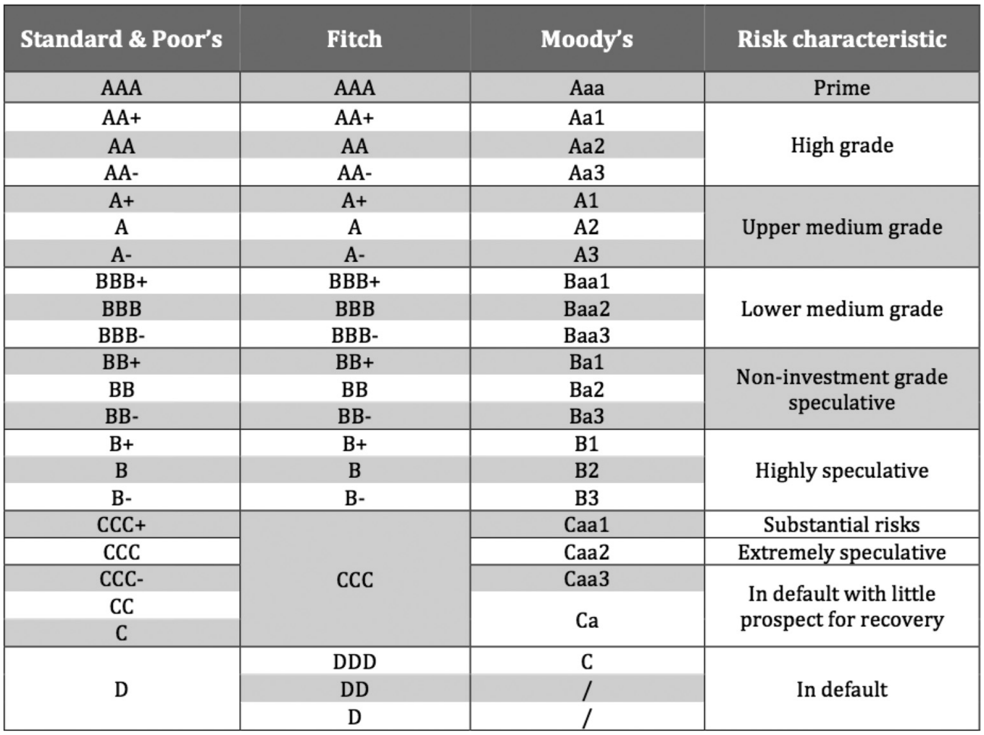
The DQVM methodology proposed in this article is focused on a specific type of process, called the *scoring process*. We define it as a process that, on the basis of given inputs, provides as output an assessment (denoted as *Process Outcome, y*) that is a ranking or a rating for an *object* under investigation on a given scale of values. As discussed in the related work, this type of assessment is typical of decision-making processes, such as in financial institutions (Moges et al. 2012). For instance, objects of investigation in a financial institution are the customers asking for credit to whom a rating is associated, which is used as a basis for decision making.

In this section, we illustrate the characteristics of this type of process using a simplified customer rating process as a running example. Such a process regards the assessment of the Probability of Default (PD) in a financial institution. All European financial institutions have to fulfill regulatory principles contained in the Basel accords[1] in order to estimate and measure the risk level addressed by the banks themselves and then to implement some strategies to manage it. Some of these regulatory principles consist of estimating credit risk parameters that include PD: the probability that a given customer in a certain time horizon is in "default" with respect to the debt that customer has with the bank. Thus, the PD allows the institution to understand if the counterpart may not be compliant with the "contract." In order to differentiate customers according to their PD, a rating class is assigned to each of them. In Figure 1, examples of such rating scales are provided by some of the major rating agencies. Each rating class is correlated with a PD interval. The best rating class (i.e., AAA according to Standard & Poor's) corresponds to the lower PD values. Vice versa, the lowest class D indicates the default situation; therefore, it is associated with a PD equal to 1.

In a financial scoring process, the rating is evaluated through a series of pertinent activities that take internal and external information as input and follow well-defined rules; the process outcome is the rating obtained as a result. The possible sequences of steps depend on different operating conditions and customer information, and can be represented using a process model to describe the sequence of steps, input and output documents, and decision points. In general, insertion order and data updates are important: in fact, a user input or a data quality control can improve the quality of previously inserted data or, vice versa, in case the quality of the source is not adequate, an update from an external source can decrease the quality of already available data for the decision.

As a running case for this article, a simplified scoring process was derived from the descriptions of several rating procedures (e.g., Lehmann (2003)). This case is represented in Figure 2 using Business Process Modeling Notation (BPMN), a widely used process modeling notation (Weske 2012). Note that using a process model does not necessarily imply automation through a workflow management system. The process model represents the different tasks that have to be executed, such as data gathering, automatic execution of mathematical operations or models, or manual activities.

[1] http://www.bis.org/publ/bcbsca.htm.



Standard & Poor's	Fitch	Moody's	Risk characteristic
AAA	AAA	Aaa	Prime
AA+	AA+	Aa1	High grade
AA	AA	Aa2	
AA-	AA-	Aa3	
A+	A+	A1	Upper medium grade
A	A	A2	
A-	A-	A3	
BBB+	BBB+	Baa1	Lower medium grade
BBB	BBB	Baa2	
BBB-	BBB-	Baa3	
BB+	BB+	Ba1	Non-investment grade speculative
BB	BB	Ba2	
BB-	BB-	Ba3	
B+	B+	B1	Highly speculative
B	B	B2	
B-	B-	B3	
CCC+	CCC	Caa1	Substantial risks
CCC		Caa2	Extremely speculative
CCC-		Caa3	In default with little prospect for recovery
CC		Ca	
C			
D	DDD	C	In default
	DD	/	
	D	/	

Fig. 1. Rating scales of the major financial services companies.

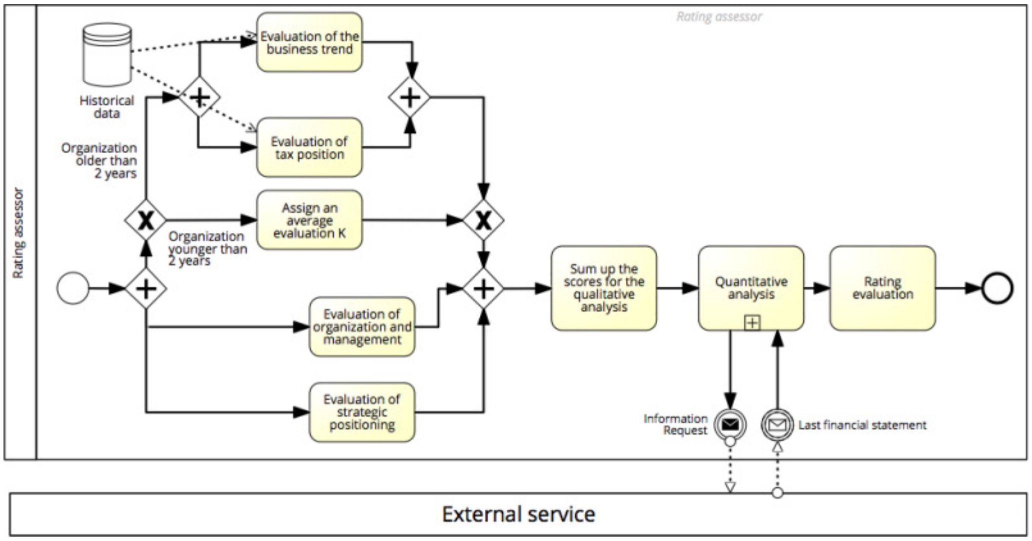

Rating assessor
Rating assessor
Historical data
Evaluation of the business trend
Evaluation of tax position
Organization older than 2 years
Assign an average evaluation K
Organization younger than 2 years
Evaluation of organization and management
Evaluation of strategic positioning
Sum up the scores for the qualitative analysis
Quantitative analysis
Rating evaluation
Information Request
Last financial statement
External service

Fig. 2. An example of the scoring process.

In our running example, the rating of the reliability of an organization is evaluated by means of a mix of qualitative and quantitative analyses. The *qualitative analysis* results from the assessment of four components:

- *Evaluation of the business trend*: On the basis of historical data, it is necessary to assess the stability of the business trend.
- *Evaluation of tax position*: It considers the debts that the analyzed organization has with public institutions.
- *Evaluation of organization and management*: It focuses on the flexibility, i.e., the capability of the organization to be adaptive within the competitive environment.
- *Evaluation of the strategic position with respect to the specific market*: This indicator combines the trend of the market and the strategic positioning of the organization.

Note that the evaluation of the business trend and the evaluation of the tax position are supposed to be automated tasks performed by a software system that considers a historical database to make the decisions. Such evaluations can be performed only if the considered organization has a history—therefore, if it is in operation for more than two years. If the organization is in operation for less than two years, the rating assessor will use a default factor K, which is the average evaluation for that specific sector of activity.

The *qualitative analysis* results from the sum of these indexes and defines an initial numerical rating.

The *quantitative analysis* considers the last financial statement (retrieved using an external service) and judges the financial position of the company by providing a second numerical rating.

The final rating (i.e., the process output) is calculated by considering the average of the qualitative and quantitative ratings, and is mapped to the S&P rating scale.

In the evaluation of the quality of the process output, the relationships between the input and the output quality are the main aspects considered in this article.

The goal of the DQVM is to analyze such relationships and to evaluate critical points that could affect the quality of the generated ratings.

In the running example, the process described in Figure 2 is enriched with DQCs as represented in Figure 3: DQC1 is inserted at the end of the qualitative assessment phase and DQC2 is inserted after the quantitative assessment phase.

In this article, we also focus on analyzing the usefulness of such controls, i.e., their sensitivity to data-quality problems and their ability to capture and take action for the most critical problems. As mentioned before, the goal is not only to assess data quality by itself but also the relationships of the effects of the quality control actions on the final result. Such an evaluation can be very useful for establishing if the quality controls in place are effective and possibly also for cost–benefit evaluations considering the effort of preventive controls on data quality and their impact on the final result.

Finally, we discuss some aspects on the evaluation of scoring processes, in particular, the way in which the processes can be improved considering the cases summarized in the table presented in Figure 4. This table refers to situations in which data are not fit for use; thus, they are characterized by a poor data-quality level for the considered process. The last column of the table provides suggestions about the proper actions to enact by considering the effect of DQ issues on the analyzed process and the existence of DQ controls. Note that if the effect of poor quality on the process outcome is low in general, then the suggestion indicated for all situations is the *laissez faire* approach. This method consists of simply reacting to issues when they occur (Redman 1996). The use of DQCs is instead suitable when the effect of poor quality is high: the introduction of

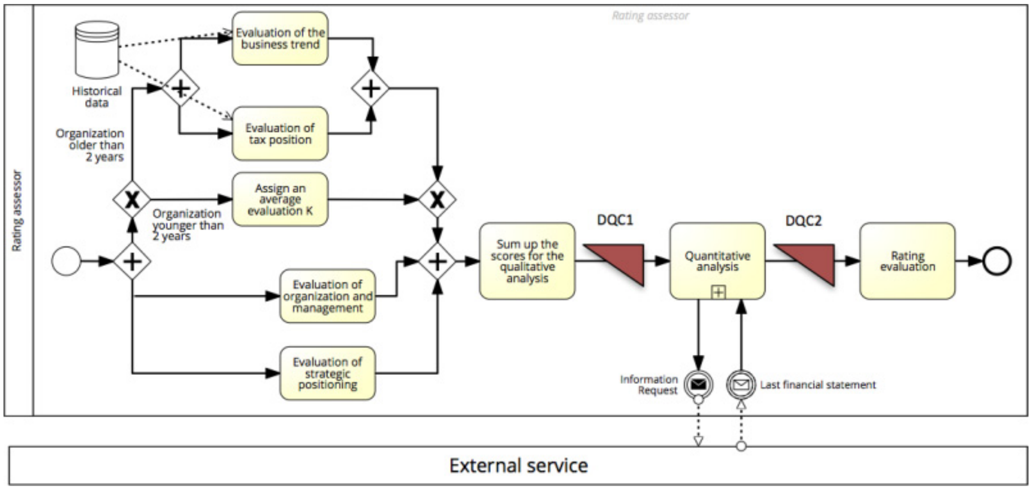
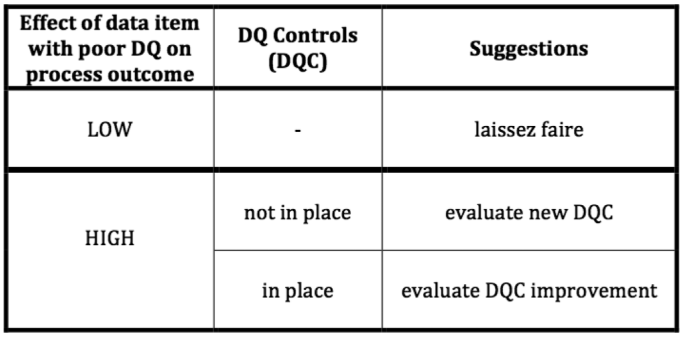

Rating assessor
Rating assessor
Historical data
Evaluation of the business trend
Evaluation of tax position
Organization older than 2 years
Organization younger than 2 years
Assign an average evaluation K
Evaluation of organization and management
Evaluation of strategic positioning
Sum up the scores for the qualitative analysis
DQC1
Quantitative analysis
DQC2
Rating evaluation
Information Request
Last financial statement
External service

Fig. 3. The example scoring process enriched with quality controls.



Effect of data item with poor DQ on process outcome	DQ Controls (DQC)	Suggestions
LOW	-	laissez faire
HIGH	not in place	evaluate new DQC
	in place	evaluate DQC improvement

Fig. 4. Suggestions for improving data quality on the basis poor quality effects.

new controls and/or the improvement of the existing ones has to be evaluated by, for example, performing a cost–benefit analysis.

What we want to assess with the methodology is the impact of data source quality on the obtained process outcome and if the existing DQCs are effective. As we show in the next section, we want to identify situations in which quality controls are not in place but needed along with those situations in which quality controls are not effective (or partially effective) and the same DQ problem is found again in different steps of the scoring process.

4 DATA QUALITY VALIDATION METHODOLOGY (DQVM)

As mentioned in the previous section, the approach proposed in this article aims to support the evaluation of the quality of the results of scoring processes, in particular, the validation of DQCs applied to a process. We assume that the target of the research is the analysis of a *Scoring Process P* that produces a final Process Output (y) and in which *m Data Quality Controls* (DQC_i) have been inserted to improve the quality of the result even when process inputs are characterized by poor DQ. The quality of the result of the process is used to identify the weights (or relevance) assigned

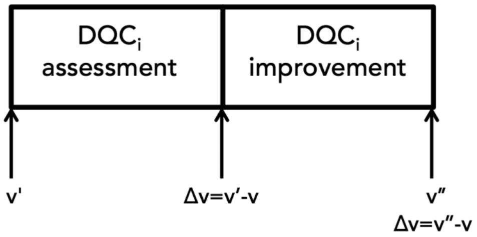


Fig. 5. Data Quality Control structure.

to *data quality dimensions DQ*, where each dimension is associated with a given relevance *weight* w_q that is proportional to its impact on the final result in case of poor quality.

The quality of the output of a process depends on the quality of different elements:

- *Input data*: Initial and inputs needed during the evaluation process.
- *Data sources*: Internal and external sources accessed during the evaluation process.
- *Generated data objects*: Data produced by a specific process task.
- *Data Quality Controls (DQC)*: Quality blocks (Shankaranarayanan et al. 2000) are inserted into the scoring process and can be viewed as being composed of two parts (Figure 5): an *assessment part* and an *improvement part*. The former is able to detect poor quality issues, while the latter is associated with rules that address them by correcting errors or notifying the presence of anomalies to the actors in charge. The effectiveness of a DQC depends on its ability both to capture DQ faults and to improve the quality of the corresponding data items.

In this scenario, we focus on the verification of the data quality of the outcome of the process and of the capability of the DQCs to detect and correct errors.

In DQVM, the DQ of the outcome of the process and the effectiveness of DQCs are verified using data fault injection techniques, as introduced in Section 2. The validation is performed by analyzing business process outputs obtained through simulations. These simulations are carried out by altering the data used in the business processes in an appropriate way (i.e., introducing DQ faults) in order to identify the situations in which poor DQ is critical for the correct process execution. From this perspective, we define as *failure* any output that differs from the one obtained in the *golden run*—in other words, correct process execution—by more than an accepted tolerance value.

In summary, DQVM aims to evaluate, in an objective way, the impact of DQ faults on the analyzed processes instrumented with DQCs in order to

- define the *importance of the different DQ dimensions* and assign them weights accordingly or verify the validity of weights assigned by domain experts;
- assess the *effectiveness of the DQCs* present in the process (if any); and
- analyze the process *behavior in the presence of abnormal situations* and its capability of managing poor DQ through its DQCs.

The DQVM is composed of the following sequential steps (Figure 6):

- *Scoring process modeling and preliminary analysis*: This step, described in Section 4.1, consists of modeling the scoring business process in order to better understand the activities that compose it and the related information flow. The desired output of this step is to identify (i) the stages that may be critical, i.e., those that are prone to the introduction of DQ

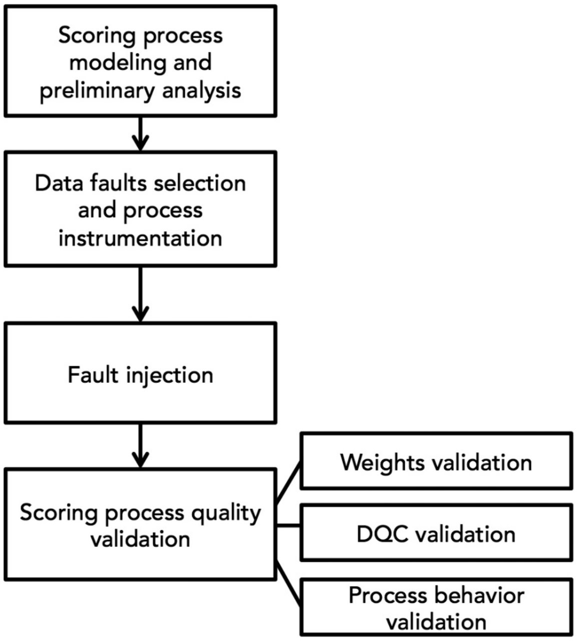

Scoring process modeling and preliminary analysis
Data faults selection and process instrumentation
Fault injection
Scoring process quality validation
Weights validation
DQC validation
Process behavior validation

Fig. 6. The DQVM steps.

faults; and (ii) the list of the process data items, whether retrieved from external sources or directly produced from the activities themselves. In this way, it is possible to identify where and on which data items the DQ faults should be placed in the fault injection step.

- *DQ fault selection and process instrumentation*: In this step, the DQ faults are designed on the basis of the DQ dimensions that we need to evaluate. At this point, the process model is enriched by adding the so called *Fault Injector Blocks*, which represent the additional activities that inject DQ faults to generate anomalous situations. This step is described in Section 4.2.
- *Faults injection*: This step includes execution of the business process simulations with injected faults. Each simulation includes the injection of one fault (Section 4.3).
- *Scoring process quality validation*: In the last step, presented in Section 5, the results of each simulation are compared with the ones obtained from the golden run in order to determine the distance between the actual result and the expected one. The analysis of these differences is the basis for evaluating the relevance of the DQ dimensions and for validating the DQCs and process behavior.

4.1 Scoring Process Modeling and Preliminary Analysis

Process modeling is usually the starting point for understanding the details and capturing critical aspects of a business process that has to be analyzed. In particular, through the process modeling step, it is possible to investigate different features of the process itself and to identify

- the *information sources* that provide input data in order to understand which data may contain errors before the process starts;
- involved *roles* in order to associate a responsible actor to each activity;

- relationships among different activities, in order to analyze the *information flow*, i.e., the sequence of operations performed on the same data item;
- *interactions with external businesses* in order to understand the involved systems and their relations with the process itself; and
- the available *DQCs* in the process and, consequently, the considered DQ dimensions for each data item in the DQC.

Note that this approach based on process modeling does not necessarily require that a process model already exists. The process might be implicit for achieving a specific output: it can be embedded in a set of formulas for analysis or in an application supporting the experts in the evaluation. Process modeling is necessary for identifying the relevant aspects in the business process, the order of execution of the activities, and the DQC actions being performed in order to make the following validation steps possible.

Two kinds of analyses are performed during the preliminary process analysis step: the dataset analysis and the process model analysis.

The **Datasets analysis** aims to understand the meaning of data involved in the analyzed business process and to determine the domain of the used data and the constraints that they have to satisfy in order to detect possible incorrect values.

The **Process model analysis** aims to analyze the process model in order to identify the possible critical points that might result in poor DQ in the process outcome. Critical situations that may emerge from process modeling correspond to typical patterns, such as the following.

- *Data retrieval from data source*: A process activity uses data retrieved from an external source. In this case, on the one hand, the information stored in data sources may contain some errors. In fact, we often do not know how the information has been stored within data sources and whether this information was previously controlled in order to verify its correctness. On the other hand, incorrect values might also arise from the IT infrastructure responsible for data transfer: the transfer operations are not correct or errors are generated (e.g., wrong interpretations of IDs or codes, temporal misalignments, and the like).
- *Message exchange*: The messages exchanged between two actors involved in the process. Also in this pattern, data can be affected in the transfer stage: the message content can be erroneously modified during the communication between actors or can be misinterpreted when different encodings or conventions are used by the communicating parties (e.g., using different rating scales).
- *Data objects used in exclusive branches of the process*: The possibility of taking different execution paths depending on the outcome of some evaluation conditions. This particular pattern leads to a very important critical aspect. In fact, it may happen that data used to decide which path has to be taken are incorrect and that, as a consequence, the decision taken is wrong.
- *Interaction with knowledge workers*: The activities in which human interaction with the system is expected. In particular, it is known that people can commit mistakes that might have a direct consequence on process execution.
- *Data generated in processing tasks*: The creation of data objects that represent a valuable data item in processing tasks or in the process output.

As a result of this step, a list of data items (including data and messages) and the points at which they are generated and used in the process is produced. Considering our example, the critical points to consider are highlighted in Figure 7.

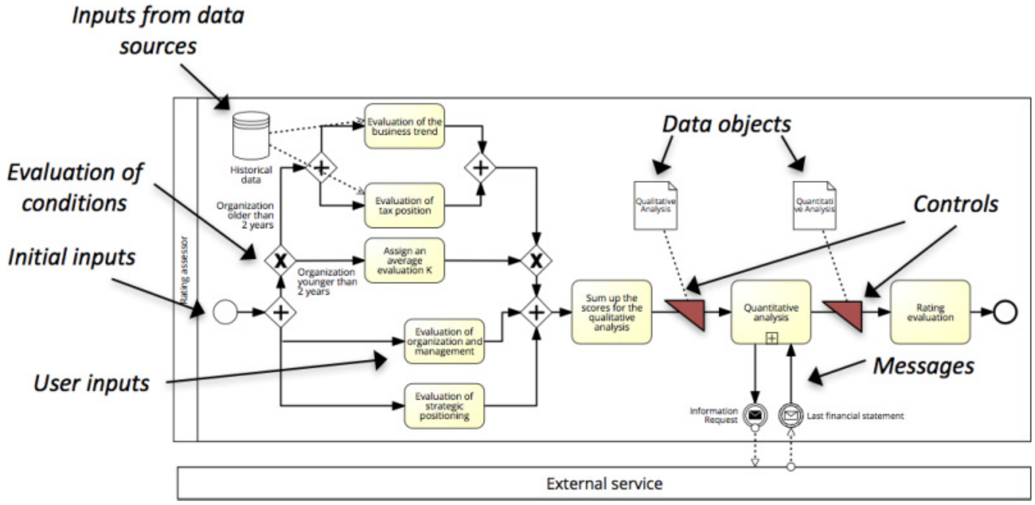

Inputs from data sources
Evaluation of conditions
Initial inputs
User inputs
Data objects
Controls
Messages
Rating assessor
Historical data
Organization older than 2 years
Organization younger than 2 years
Evaluation of the business trend
Evaluation of tax position
Assign an average evaluation K
Evaluation of organization and management
Evaluation of strategic positioning
Sum up the scores for the qualitative analysis
Qualitative Analysis
Quantitative Analysis
Quantitative analysis
Rating evaluation
Information Request
Last financial statement
External service

Fig. 7. Critical points and related data items in the example scoring process.

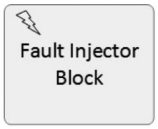


Fig. 8. Fault Injector Block representation.

4.2 Data Fault Selection and Process Instrumentation

For performing the DQ validation on a given scoring process, we assume that a set of correct executions of the process are available. Each correct execution (i.e., *golden run*) results in a correct process output.

In our method, we introduce systematic perturbations on such executions, introducing data faults, with the goal of assessing their impact. Data faults correspond to data variations that are potentially able to cause process failures or inefficiencies.

4.2.1 Process Instrumentation. The data faults have to be placed inside the process at suitable points. To model the perturbation more efficiently, we propose a modeling extension: a new task, called *Fault Injector Block*, that perturbs data inside the process, generating data faults. For example, using the BPMN notation, the symbol associated with this extension can be the one shown in Figure 8, consisting of a task marked with a flash of lightning. It is important to remark that this block has to be inserted in the activity flow and that it needs to be properly configured to inject the faults for which it is prepared. We assume that a fault injector block can generate one or more types of faults, and that each fault is associated with only one quality dimension.

The positioning of fault injector blocks is a crucial aspect of the methodology. In fact, placing these blocks correctly allows them to inject faults into a business process in an appropriate manner. In our approach, we position the blocks each time that we have a pattern among the ones listed in the previous section. Therefore, we assume associating a fault injector block to each task of the process that is receiving either external data (from data sources or from human input) or an external message.

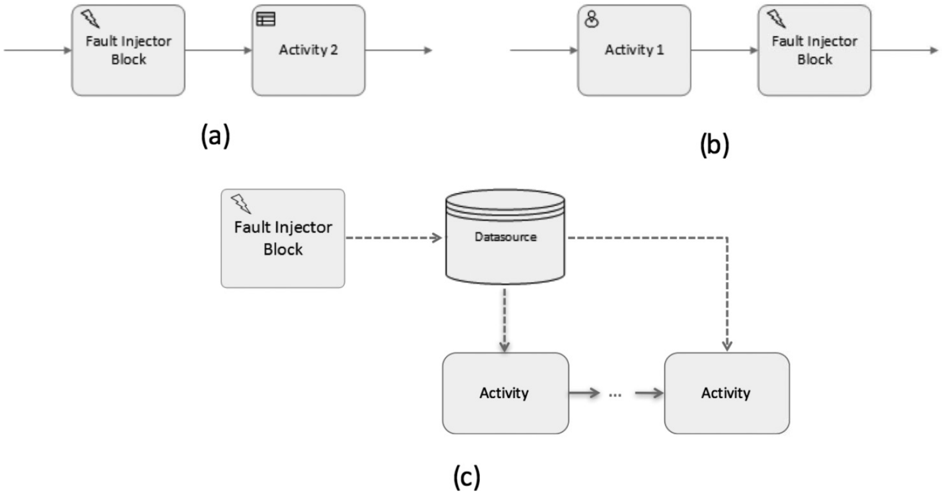


Fig. 9. Injection patterns.

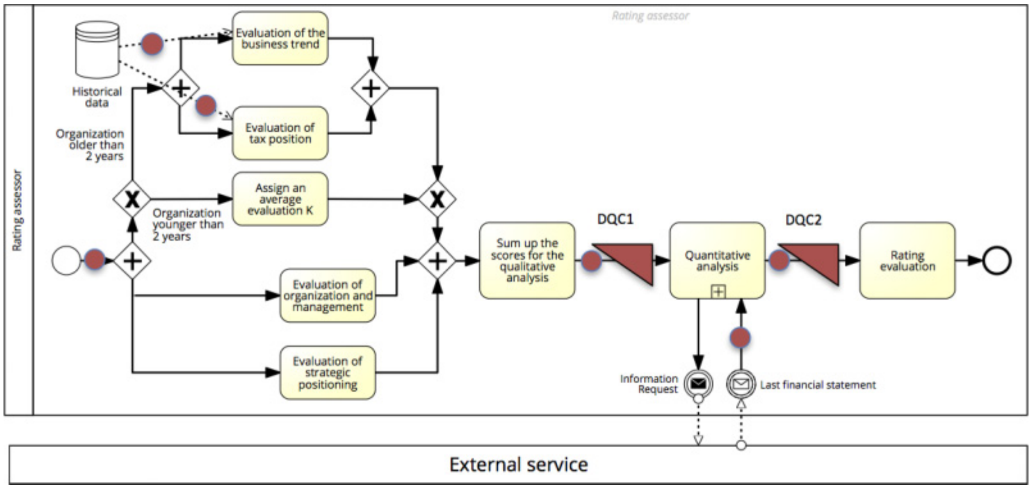

Rating assessor
Rating assessor
Historical data
Evaluation of the business trend
Evaluation of tax position
Organization older than 2 years
Assign an average evaluation K
Organization younger than 2 years
Evaluation of organization and management
Evaluation of strategic positioning
Sum up the scores for the qualitative analysis
DQC1
Quantitative analysis
DQC2
Rating evaluation
Information Request
Last financial statement
External service

Fig. 10. Points in which faults injector blocks have to be placed.

Figure 9 shows the main patterns used for modifying the process for injecting faults. The first two patterns are used to modify data received by an activity (pattern a) or produced by an activity (pattern b), respectively. Pattern b is mainly used when the data produced by an activity are the input for an exclusive branch to evaluate a condition when a data object is created. Pattern c shows how data sources, which are used in several activities in the process—i.e., shared data sources—are modified using a fault injector block.

The process in our example can be instrumented as represented in Figure 10.

It is worth noticing that the fault injector blocks can be activated or not. In case a fault is not activated during the execution of a process instance, the corresponding fault injection block acts as a "transparent" component without modifying the process behavior in any way. In this manner, only one fault at a time can be activated and tested in the simulations, while the process is instrumented with all possible fault injection blocks.

4.2.2 Data Faults Selection. Once the points in which faults have to be injected have been defined, it is necessary to identify the data items to be perturbed through the insertion of faults and the associated DQ dimensions that have to be evaluated. In fact, the selection of DQ dimensions drives the choice of the types of faults that have to be injected to test the process. The data items are selected via the process analysis performed in the previous step, considering the list of data items for the process. For each data item, the possible DQ faults must be generated. In the following, we consider for our running example three of the most used DQ dimensions, i.e., accuracy, coherence, and completeness. Other dimensions could be considered as well, provided that they can be injected into the process through an appropriate Fault injection block.

Looking at their definitions, such dimensions are mainly affected by two types of faults (Fugini et al. 2009):

- *Value mismatch* that can be caused by typos, different formats, semantic conflicts, and delays in update operations
- *Missing data* due to value unavailability

Therefore, to introduce faults affecting the quality of data along these dimensions, we use data update and data deletion operations. With *data update* operations, data are changed by introducing some typos or substituting values. The resulting incorrect values may or may not be within the acceptable ranges for the data being considered, possibly generating different consequences on process execution. *Data deletion* operations delete values previously assigned to given variables instead.

Each type of data fault can be generated with different degrees of perturbation: in the case of numerical values, for example, values can be derived from the correct ones in different percentages of modification, as considered relevant in the given application.

4.3 Fault Injection

In this step, the process is executed by systematically injecting all the designed faults. We assume that a set of correct process executions exist as a test set, considered as *golden runs* for the process. The fault injection process is applied to the process test set; for each process execution, one fault is injected at a time, thus executing the process as many times as required to inject all the faults one by one. The results of the faulty executions have to be compared with the results obtained from the considered golden run: such a comparison shows the impact of the different faults on the process output and is discussed in next section.

4.3.1 Data Fault Injection System. In order to perform the simulations needed for using the DQVM, we designed the architecture of a simulation system able to automatically execute the process and inject faults. This system aims to faithfully replicate the business specifications with which real processes are implemented. An example of implementation of this architecture in a real case study is described in Section 6.

Once the process model is available, it can be easily coded to be automatically executed. Note that for the processes that are already automated, the implementation of a simulator is easier since there is already an original source code that can be reused. In any case, the simulation system must be able to inject faults in order to be used effectively for the methodology.

The functional architecture of the data fault injection system we developed is represented in Figure 11. The system operations are managed by the *Controller* that is responsible for starting both the simulations and the analysis. More specifically, the Controller gathers the faults to consider from the *Fault Repository* and properly configures the *Data Fault Injector Blocks*. The simulations are performed using a *Process Simulator*, which corresponds to a replica of the business process

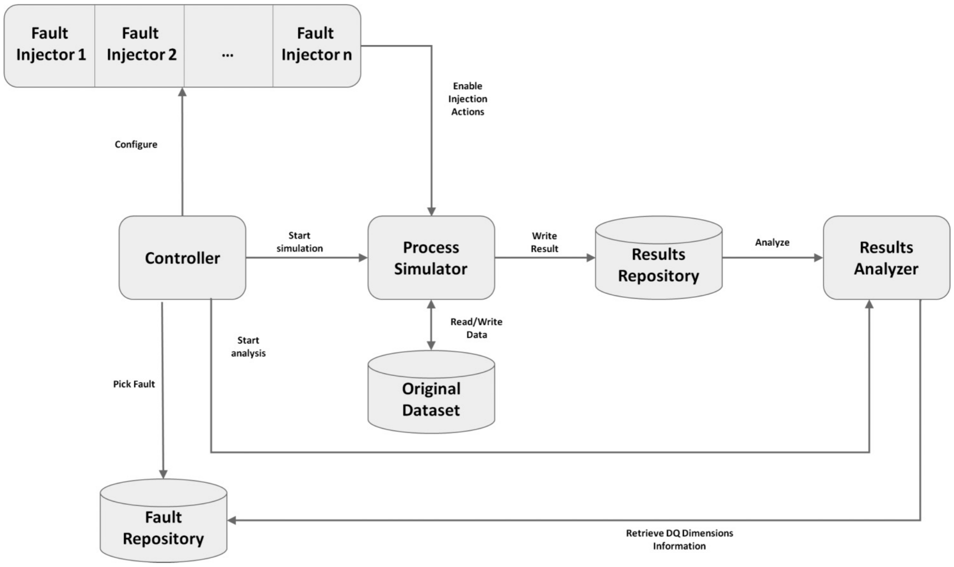


Fig. 11. Functional architecture of data fault injection system.

under examination. This replica is also enriched by the injection actions provided by the *Fault Injector Blocks*. The controller manages activation of the fault injectors: for each fault contained in the *Fault Repository* a series of corresponding process faulty executions are launched and their results are stored in the *Results Repository*. Such a repository will contain the data needed for the result analysis phase that has to be started by the *Controller* once all the possible simulations have been executed.

5 SCORING PROCESS QUALITY VALIDATION

In this section, we illustrate the core of the DQVM methodology, i.e., the analysis of the fault injection results.

This analysis leads to different contributions:

- The validation of the *importance given to DQ dimensions* (Section 5.1)
- The *validation of the DQCs* (Section 5.2)
- The *identification of anomalies in process behavior* (Section 5.3)

Before providing details about such contributions, it is necessary to clarify that data faults may affect the outcome of the processes in different ways. In fact, as the process is already enriched with DQCs, the impact of data faults may be little or none even when data faults are present. In the following, we will compare the output of the process y_k when the kth fault is injected and the output of the golden run process y. For the purpose of this article, we classify faults as *transparent faults*, *light faults*, and *heavy faults*. We define *transparent faults* as the faults that leave the process output unchanged: the output of the faulty execution is equal to the output of the golden run (i.e., $y_k = y$). *Light faults* refer to faults that have a small impact on the output. In particular, we define a threshold δ for the variation associated with a significant impact on the process output. Light faults are those faults for which the following conditions hold:

$$| y_k - y |= e_k \quad and \quad 0 < e_k \leq \delta,$$

where e_k is the variation (*error*) on the output rating y due to fault k.

Heavy faults are those faults for which the difference between the expected and the actual value is greater than δ:

$$| \, y_k - y \, | = e_k \quad and \quad e_k > \delta.$$

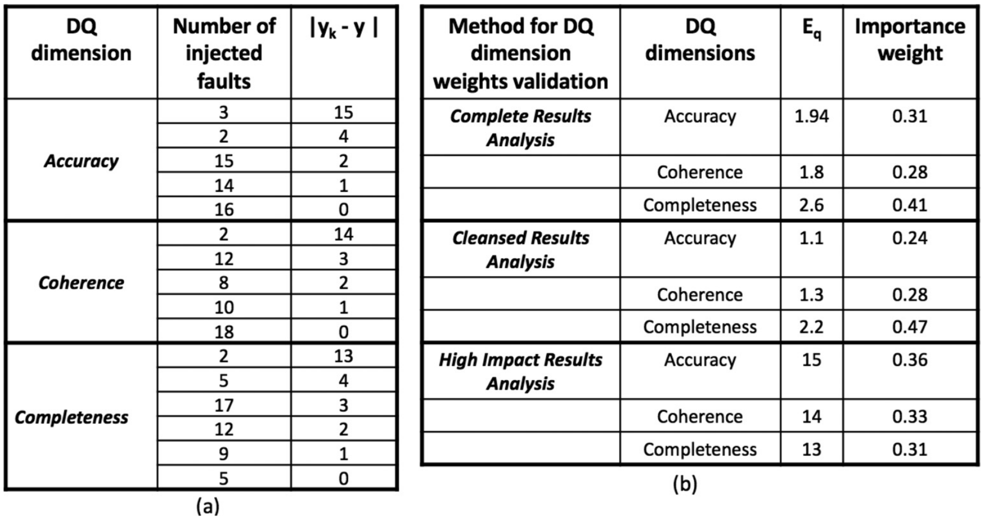
The assessment of e_k depends on the type of process output. In fact, if the output y is a numeric value, e_k results from the absolute difference between y_k and y; if the output is a string, the difference can be computed by adopting some algorithms to evaluate similarity among strings (e.g., it is possible to consider the methods listed in Batini and Scannapieco (2016)). Finally, the output can be a string belonging to a finite domain (such as in the case of the rating classes): in this case, we use a function $r(y)$ to map the string to a numeric domain. For example, let us consider the S&P classification. It has 22 classes and we can associate a number with each of them, e.g., $r(AAA) = 22, r(AA+) = 21 \ldots r(D) = 1$. With this mapping, we can estimate e_k as the absolute difference between y_k and y. In this scenario, we consider that a transparent fault does not cause changes in the rating class, a light fault can cause a small change in the rating class (i.e., $\delta = 1$, that is a previous or subsequent rating class, e.g., from A+ to A), and a heavy fault is responsible for changing the rating class by two or more positions (i.e., $e_k \geq 2$, e.g., from A- to BBB-).

5.1 Validation of the Relevance Given to Data Quality Dimensions

In general, DQ dimensions are characterized by weights that are used to express their relevance in the considered business context. These weights are typically defined by DQ experts in a subjective way on the basis of their expertise. However, since the knowledge workers often do not have tools and objective evaluations that allow them to discriminate between the individual DQ dimensions, in most cases, the weights are set to be the same for all dimensions. As a result, without knowing which DQ dimension has to be monitored in depth, the common behavior is to adopt improvement techniques for guaranteeing generic DQCs for each dimension. Direct consequences of this approach are (i) possible problems caused by deficiencies in terms of the number and quality of controls related to the most relevant dimensions in the considered context or, on the contrary, (ii) a possible waste of resources due to an overestimation of the number and type of controls needed for less important dimensions in the context under examination. Therefore, knowing which are the most important DQ dimensions or, more generally, which is the relative importance of each dimension with respect to the other ones allows organizations to better design and place DQCs. In order to understand if the assigned weights correspond to the true relevance of the DQ dimension in the considered process, the DQVM proposed in this article can be used to validate the choices made by DQ experts. In particular, the methodology is able to highlight the situations in which the relevance of some dimensions have been underestimated and no proper DQCs have been designed.

Evaluating the differences between the outputs of the faulty business process executions and the output of the golden run executions is the first step in validating the relevance weights given to the different DQ dimensions. The errors e_k are divided into clusters on the basis of the DQ dimensions affected by the different faults. Considering, in fact, that each kth fault has an impact on the qth DQ dimension, we can reclassify all the different e_k values as e_{qk} values in order to clarify the DQ dimension on which the fault has impact. We can compute an aggregate indicator E_q according to three different approaches:

- *Complete result analysis*
  In this approach, the E_q aggregate indicator is computed as the algebraic average of e_{qk} values relative to each of the q dimensions.
- *Cleansed result analysis*
  In this approach, the simulation results are cleansed from the outliers—i.e., values deviating from the average results of the simulation—in order to determinate the E_q aggregate



DQ dimension	Number of injected faults	$\|y_k - y\|$
Accuracy	3	15
	2	4
	15	2
	14	1
	16	0
Coherence	2	14
	12	3
	8	2
	10	1
	18	0
Completeness	2	13
	5	4
	17	3
	12	2
	9	1
	5	0

(a)

Method for DQ dimension weights validation	DQ dimensions	E_q	Importance weight
Complete Results Analysis	Accuracy	1.94	0.31
	Coherence	1.8	0.28
	Completeness	2.6	0.41
Cleansed Results Analysis	Accuracy	1.1	0.24
	Coherence	1.3	0.28
	Completeness	2.2	0.47
High Impact Results Analysis	Accuracy	15	0.36
	Coherence	14	0.33
	Completeness	13	0.31

(b)

Fig. 12. Application of DQVM for the weight validation.

indicator value. One possible method used to delete the outliers is Winsorization (Hastings et al. 1947), which consists of cutting values above the 95th percentile, to eliminate samples associated with faults characterized by a very high impact, and those below the 5th percentile, to eliminate samples associated with faults characterized by a null or almost null impact. After having cleansed the analyzed sample and clustered fault results according to the affected DQ dimension, the E_q indicator is computed as the algebraic average of the cleaned set of e_{qk} values. Note that the cleansed result analysis method is useful for gaining an unpolarized result analysis, i.e., not conditioned by some particular results. In addition, it can also be seen as an outlier detection phase that aims to manage separately "extreme" result values, i.e., those that deviate from the average simulation results.

- *Highest impact results analysis*
  In this approach, the E_q aggregate indicator value is the maximum e_{qk} for each qth dimension in order to consider only the execution with the greatest impact on each dimension. Contrary to the previous case, with this approach, the impact of the consequences of outliers is considered to be the most effective in assessing the relative importance of the DQ dimensions.

Hence, the relative relevance of each DQ dimension, called w_q, is computed as the ratio of the aggregate indicator referred to the qth dimension (E_q) and the sum of the aggregated indicators. As a result, it can be calculated using the following formula:

$$w_q = \frac{E_q}{\sum_{q=1}^{Q} E_q}.$$

The obtained weights provide an objective basis for evaluating the real importance of the different DQ dimensions: the greater the deviation from the expected output, the greater the relevance of the DQ dimension.

In order to clarify the approach, let us consider our running example in one instance of execution of the process and let us assume to have injected 50 faults into the process variables for each of the three quality dimensions being considered. Figure 12 presents the results we obtain.

Figure 12(a) illustrates, for each dimension, the different results obtained (i.e., $|y_k - y|$ values) and the corresponding number of injected faults that caused such results. Figure 12(b) results from application of the methods described above. We calculated E_q and the relevance weights for the different dimensions applying the three approaches listed above. Note that the importance given to dimensions (i.e., weights) depends on the applied method.

While for illustration purposes we show just the application of all faults to one instance, in general, this analysis process is applied to a set of instances (defined before as the golden runs for the process), evaluating the average effect of each fault on all considered cases.

Each result y'_k corresponds to the average, computed on the considered set, of the absolute variation of the results with regard to the golden run due to fault k, which is modeled as

$$|y'_k - y| = \frac{\sum_{i=1}^{n} |y'_{ki} - y|}{n},$$

where n is the cardinality of the golden run.

This formula is used as an indicator to give a quantitative measure of the impact of each data fault on the considered set, i.e., the average error being introduced.

5.2 Data Quality Control Validation

The DQVM can also be used as an objective and useful tool for validating individual DQCs already existing within the process. In particular, as shown in Figure 5, each DQC assesses the relevant process variables with reference to the golden run of the process. As discussed in Figure 13, the impact of poor DQ on the result of the process can be the starting point for the analysis. In this section, we focus on existing DQCs and analyze their effectiveness after their execution and considering the differences in the final outcome of the execution of the process.

In the analysis, we adopt the following notation:

- m is the number of DQCs applied to the analyzed business process
- v is the vector containing the variables of each DQC during the execution of the fault-free process (golden run):

$$v = \begin{bmatrix} v_{DQC1} \\ v_{DQC2} \\ \cdots \\ v_{DQCj} \\ \cdots \\ v_{DQCm} \end{bmatrix}$$

- v' is the vector containing the output values of each DQC (where each output value can itself be a vector of values for all the considered data items in the DQC) during execution of the process with one fault injected:

$$v' = \begin{bmatrix} v'_{DQC1} \\ v'_{DQC2} \\ \cdots \\ v'_{DQCj} \\ \cdots \\ v'_{DQCm} \end{bmatrix}.$$

The desirable situation is that, among the controls, a monitoring control $control_{j^*}$ exists for the fault being considered, aimed at capturing the anomaly generated by the fault and taking appropriate improvement actions. To verify whether this happens, the $\triangle V$ vector is computed as follows:

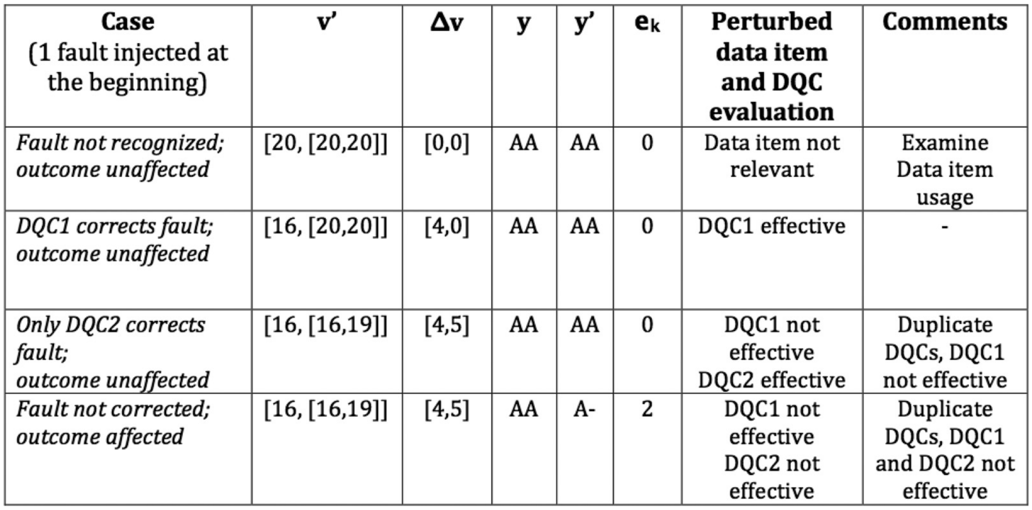
$$\triangle V = v' - v = \begin{bmatrix} v'_{DQC1} - v_{DQC1} \\ v'_{DQC2} - v_{DQC2} \\ \cdots \\ v'_{DQCj} - v_{DQCj} \\ \cdots \\ v'_{DQCm} - v_{DQCm} \end{bmatrix},$$

where differences are computed on the basis of the data types as previously discussed.

At this point, three different situations may arise, depending on the $\triangle V$ vector values:

- The $\triangle V$ vector has only *one value different from zero*. This means that, as is desirable, there is only one control that monitors the consequences of the fault inserted in the process. It can therefore be stated that the control affected by the fault is independent from the other ones: in practice, these controls are valid from the point of view of the fault taken into account. When the fault is identified, actions are taken in the scoring process so that the following DQCs are not affected during the rest of the scoring process.
- The $\triangle V$ vector is characterized by *more than one value different from zero*, i.e., depending on changes due to fault inclusion. The occurrence of such a situation is indicative of the fact that there are duplicate controls currently implemented or that these controls have been defined too roughly and the actions taken in the improvement phase are not sufficient. In this case, starting from the introduced anomaly analysis, the indication is that there is a need to refine these generic controls or to evaluate if the removal of duplicated controls to capture the same anomaly is possible.
- The $\triangle V$ vector is null, i.e., the fault presence *does not originate any variation* in the implemented controls. This means that the controls are not able to detect the faults and, if $y \neq y'$, a new control is needed to monitor the anomalous situations associated with the introduced fault.

In order to better explain this procedure let us consider the example represented in Figure 10. Let us assume that we aim to check the effectiveness of the DQC1 that is situated before the Quantitative Analysis activity. We injected a fault that deleted the evaluation of the tax position introducing data incompleteness issues. DQC1 was designed to check whether data are complete; if missing values occur, it sends an alert. Missing values should be completed with proper values in the improvement part of the control. The considered process has two quality controls, DQC1 and DQC2, and the v vector derived from the golden run is $v = [20, [20, 20]]$, where 20 is the value of the data item of the first quality control and represents the sum of the scores of the qualitative analysis, while [20,20] is the value of the second quality control and represents the values for the data items of the quantitative and the qualitative analysis, respectively. The final rating associated with the enterprise in the golden run; i.e., the final Process Outcome y, is AA. In Figure 13, we discuss some possible scenarios after execution of the injected fault. In the first row, the fault is not recognized by the DQCs, but the result is not affected. In this case, the relevance of the considered data item in which the fault is injected (tax evaluation) is questioned, and its relevance has to be checked. In the second row, an execution scenario in which the first control DQC1 assesses and improves the quality of the faulty data is illustrated, with no effect on the process outcome. The last two rows illustrate scenarios in which DQC1 is not effective: in the first case (third row), DQC2 is effective both in assessing the fault and improving the data quality and the final outcome of the



Case (1 fault injected at the beginning)	v'	Δv	y	y'	e_k	Perturbed data item and DQC evaluation	Comments
Fault not recognized; outcome unaffected	[20, [20,20]]	[0,0]	AA	AA	0	Data item not relevant	Examine Data item usage
DQC1 corrects fault; outcome unaffected	[16, [20,20]]	[4,0]	AA	AA	0	DQC1 effective	-
Only DQC2 corrects fault; *outcome unaffected*	[16, [16,19]]	[4,5]	AA	AA	0	DQC1 not effective DQC2 effective	Duplicate DQCs, DQC1 not effective
Fault not corrected; outcome affected	[16, [16,19]]	[4,5]	AA	A-	2	DQC1 not effective DQC2 not effective	Duplicate DQCs, DQC1 and DQC2 not effective

Fig. 13. Discussion of the analysis of the running example with one injected fault.

process is still correct; in the last row, both DQCs fail to improve the data quality and the final rating is heavily affected.

This example shows the role of the DQCs: as discussed in Figure 5, a DQC does not only assess but it also contributes to improving the quality of data; thus, even with faults, the process is able to reach correct results if the improvement part of the control is effective (e.g., the final result is the same as that of the golden run, thanks to the action of the quality control).

5.3 Process Behavior Validation

As discussed above, a process with data faults, if designed with the appropriate quality control actions and with an appropriate set of tasks, may yield acceptable results even when data faults are present.

Process behaviors validation consists of verifying whether the process behavior is suitable even in the presence of abnormal situations, i.e., the process behavior can be considered correct if its result is equal to that of the golden run or possibly within an allowed tolerance interval δ defined for its results.

To this purpose, for each golden run, we define the extreme bounds y_{min}, y_{max} of the confidence interval for output $[y_{min}, y_{max}]$, where $y_{min} = y - \delta$ and $y_{max} = y + \delta$, beyond which a given process output has to be considered anomalous.

In fact, a process is not considered anomalous with respect to its specification if the following relation holds for each of the results y' deriving from application of the data faults defined in relation to the process itself:

$$y_{min} \leq y' \leq y_{max}.$$

If such a relation does not hold, the result of the process when data faults are present is significantly affected by the faults and we are obviously in the presence of anomalies that should be investigated in order to understand their causes. As indicated in Figure 13, in this case, the effect of poor DQ is high and an evaluation has to be performed on which are the causes of the high impact of the fault. As we were discussing in the previous section, the effectiveness of DQC can be systematically evaluated. In this section, we consider mainly the case in which DQCs are not in place. As discussed previously in Section 3, in this case, we have to evaluate whether a new DQC can be introduced in the process to assess the fault and perform an improvement action.

There are two possible situations that have to be considered:

- The introduction of a *new DQC*: This solution can be proposed when a local assessment of the DQ is possible and the error can be corrected locally.
- *Revision of the process*: This solution is proposed when the fault affects a specific branch of the process; this can occur when the faulty data item is used as a condition in a branch in the process and the branch is used to perform tasks that are considering specific faulty values, such as data out of range or missing. In this case, a DQC is not sufficient, and one or more tasks have to be performed to compensate for planned faulty data management in the process. In this case, the DVQM acts as a support to identify possible areas of the processes or subprocesses to be revised.

As mentioned for the weights analysis, the deviation from the golden run introduced by a fault is calculated on all instances of the golden run.

If there is a systematic deviation from the acceptable behavior, the process behavior with respect to the given fault presents an anomaly. This analysis, combined with the analysis of DQCs illustrated in the next section, can indicate weak points in the process and its associated quality controls.

6 CASE STUDY: A RISK MANAGEMENT SCORING PROCESS

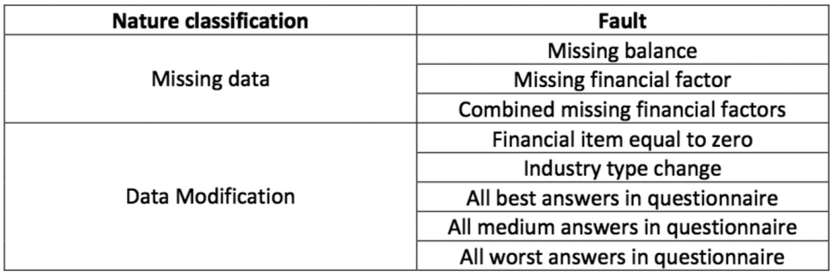
In this section, we discuss a case study applying the DQVM in a financial institution. Within the Risk Management area of the institution, different processes are adopted for evaluating different types of bank customers in terms of PD according to the internal rating–based approach as defined by the Basel Committee on Banking Supervision[2]; in this case study, we considered the process related to the evaluation of multinational counterparts. Note that the process considered earlier in the article as a running example in Section 3 and Figure 2 contains only some simplified examples of the elements present in the real evaluation processes used by the considered financial institution, which is far more complex.

The considered process uses a large amount of information in order to obtain the PD of each counterpart. In particular, 22 data items are considered in the evaluations.

The process, which is characterized by a series of formulas and parameters, consists of a quantitative module and a qualitative module. The former uses balance sheet data to obtain a quantitative evaluation, while the latter is based on qualitative questionnaires with a resulting qualitative score. All questionnaires are compiled by institution experts using the “four-eyes” principle (i.e., all questionnaires are compiled independently by two different experts). Moreover, the process is characterized by interaction with users and several data sources. The process consists of 11 subprocesses, has 3 data sources, and there are 3 involved user roles. Full details of the real process cannot be published due to confidentiality reasons and due to its highly complex structure.

The case study is based on a test set of 2,559 counterparts corresponding to 2,559 golden run evaluations. The result of the process is a PD evaluation similar to the case shown in the running example. For each counterpart, both assessment questionnaires and external data are available. The questionnaires are compiled by different internal experts based on available information and assessment rules and guidelines.

[2] https://www.bis.org/.



Nature classification	Fault
Missing data	Missing balance
	Missing financial factor
	Combined missing financial factors
Data Modification	Financial item equal to zero
	Industry type change
	All best answers in questionnaire
	All medium answers in questionnaire
	All worst answers in questionnaire

Fig. 14. Classification of injected faults according to their nature approach.

The process is also enriched with DQCs for evaluating the quality of the information used in the computation of credit parameters in several points of the process. In particular, such controls aim to assess the suitability of the quality level of data used during computation.

The process is subject to quality control procedures and quality reports are periodically produced, as required by the Basel regulations. The existing DQCs that cover all data items and all activities in the process produce high-quality evaluations of the probability of default.

In this study, we focus on the following DQ dimensions: completeness, accuracy, and coherence (consistency). These dimensions, each with an associated relevance weight, concur to determine the assessment of the overall DQ level of the evaluations performed by the financial institution. All quality dimensions are initially considered equally relevant by the experts.

The aim of the study was to

- validate the relevance weights associated to the dimensions being considered (initially all considered at the same level);
- assess the quality of the DQCs themselves, i.e., their ability to monitor critical situations; and
- identify abnormal behaviors, in which poorer quality corresponds to better evaluations in the results of the process.

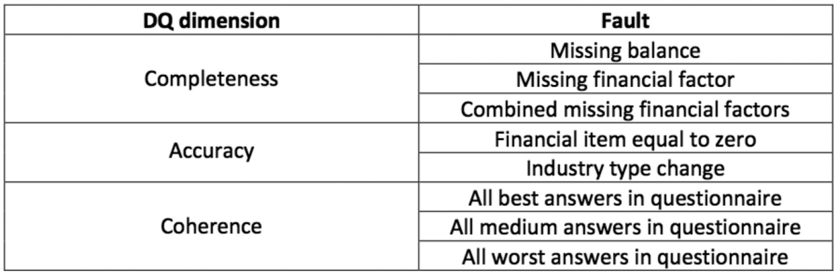
6.1 Process Modeling and Fault Injection Steps

In the first step, we modeled the scoring process using BPMN[3]. In the second step, we selected the faults to be analyzed using fault injection, based on the analysis of used data.

Relevant data items were identified using (i) the existing data dictionary, which also provided information about the valid domains for data items; and (ii) the existing DQC in the process which, even if defined in an expert-based way, allowed the identification of some possible abnormal situations already monitored by controls themselves.

In the following step, we created 150 types of faults on these variables (with an average of 6.8 faults per variable), creating both missing data faults and faults obtained modifying data. In Figure 14, we illustrate the types of faults due to missing data or modifications of data that were considered. In the case of missing data, a fault is introduced eliminating available data from the golden run. For data modifications, systematic modifications are introduced, considering modified values (a) within the allowed range of values, varying the golden-run values with different percentages (for numeric values) or with alternative values (for text values, considering both small changes such as typos, and larger modifications, such as alternative values); (b) at the extremes of the allowed range; and (c) outside the allowed range.

[3] http://www.bpmn.org.



DQ dimension	Fault
Completeness	Missing balance
	Missing financial factor
	Combined missing financial factors
Accuracy	Financial item equal to zero
	Industry type change
Coherence	All best answers in questionnaire
	All medium answers in questionnaire
	All worst answers in questionnaire

Fig. 15. Classification of injected faults according to data quality dimensions.

The following strategies are used to create the faults in the case study:

- Insertion of incompleteness (e.g., missing balance sheets, missing financial factors). The fault is obtained deleting values existing in the golden run. Combined missing financial factors were also considered.
- Setting financial items with limit values. In particular, for financial data, the zero value is considered and usually treated in a particular way, i.e., it can lead to conditions such that particular actions are taken. This fault was generated for all data items (financial item equal to zero).
- Industry type modification: In this case, the change of classification can introduce errors in the process, as different processing rules are applied for different industry types.
- Modifications to answers in the questionnaire, setting all values to best (or medium, or worst) for all answers.
- Modifications to insert incoherence between questionnaire answers. Through this fault we wanted to model conflicting situations in questionnaire answers used for qualitative score computation.

In Figure 15, faults are classified according to DQ dimensions into completeness, accuracy, and coherence fault types.

It has be noted that the classification along DQ dimensions focuses on the principal DQ issue introduced in the data as fault. For instance, incompleteness could also be considered an accuracy problem; however, if data are missing, we consider it to be an incompleteness problem rather than an accuracy problem, as this is the originating cause for DQ problems.

In addition, a DQ problem could cause multiple effects. As presented before, we focus on the global effect on the process, considering its outcome. The process outcome may be affected by a combination of possible multiple effects but also by DQCs introduced in the process with the specific purpose of reducing the impact of DQ problems. The DQVM provides evaluations based on the final outcome of the process, in which all of these effects are combined in the derivation of the final result.

We systematically generated 150 types of faults for the 22 data items based on introducing the types of faults illustrated above. The generated faults were evaluated by the experts in the financial institution: the experts identified 74 types of faults as significant for the analysis, discarding the remaining generated faults as unlikely to occur in the process execution. The selected faults included 29 accuracy faults (including values set to nulls and changed string values), 13 completeness faults (missing values), and 32 coherence values (all related to data inserted in questionnaires).

The 74 faults were introduced one fault at a time on each golden run to evaluate their effects, for a total of almost 190,000 executions of the simulation.

	Coherence	Completeness	Accuracy
Initial assessment	33	33	33
Highest impact results analysis	66	22	12
Complete results analysis	48	44	7
Cleansed results analysis	45	50	5
Final assessment (experts' evaluation)	50	40	10

Fig. 16. Comparison of the analysis approaches (percentages).

The tool for fault injection and analysis was developed on the SAS Business Analytics Framework[4], version 9.3, using the SAS programming language to create the data fault injection system for the process being considered.

6.2 Analysis of the Results

6.2.1 Analysis of Weights Assigned to the Quality Dimensions. The table shown in Figure 16 shows how the importance associated to the DQ dimensions varies applying the different evaluation methods illustrated in Section 5.

Initially, all considered DQ dimensions were given the same importance by the institution (shown as 33% in the table in the first row, indicating the initial assessment).

We show in the table the results applying the three proposed evaluation methods for the effects of the faults: highest impact, complete, and cleansed results analysis.

All evaluation methods show that the DQ dimension that originates the highest impact is coherence. In particular, considering the highest impact, coherence becomes by far the most important dimension, as the impact of lack of coherence causes the largest variations.

The table also shows that the importance of accuracy was largely overestimated. In fact, the results show that accuracy issues in data are likely to be captured by DQCs; therefore, the impact of accuracy should be considered lower in the final assessment of the quality of the result.

The table also shows that if outliers are not taken into consideration, coherence and completeness come to have a similar importance level.

In the table, we also show how the analyses yield a revised importance given to the DQ dimensions by the experts. Starting from the results of the evaluation and with the support of the DQ experts of the financial institution, who could also provide some considerations on the relevance of timeliness with their experience, new weights were proposed for DQ dimensions.

The revision of the weights (shown in the last row of the table) is also the basis for further refining the design of the existing system, increasing the attention on coherence and completeness.

6.2.2 Analysis of Data Quality Controls. Concerning the validation of controls, the validations performed with the proposed method showed that the current implemented controls are appropriately defined in order to monitor all possible critical situations that may occur in the PD computation process.

In the PD scoring process, a set of controls was introduced into the process for each DQ dimension in order to capture data DQ issues on the data used in the evaluation. These controls were established by DQ experts in the organization. The data fault injection methodology application makes it possible to have an objective evaluation of the adequacy of DQCs. According to the process described in Section 5.2, the vector v' for each implemented fault was considered in order to assess the capability of the DQC in assessing poor DQ and its ability to improve the quality of the data evaluated in terms of impact on the process output. From a comparison of all these

[4] http://www.sas.com.

vectors with the ones associated with the golden runs, we can see that, for each fault, there is only one associated vector element that varies and that the obtained score is no more distant that one level from the scoring level of the golden run, on average. This is a good indication that the DQCs are independent from each other and that they are characterized by sufficient granularity, which guarantees good quality improvement.

6.2.3 Process Behavior Analysis. In the case study, only one injected fault in the test set resulted in ratings that were out of the established bounds for the process outcome, on average.

This case was analyzed in detail: in this case, a new DQC is not a possible solution, as the handling of that type of faulty data requires complex evaluations and a branch of the process handling incompleteness was identified for improvement. In fact, changing situations may require adjustments to the way that financial situations are evaluated; therefore, previous validation tasks may change. As such, the approach is useful in identifying and validating changes in scoring processes over time and their consistency with respect to available data.

7 CONCLUDING REMARKS

The methodology proposed in this article aims to provide support for the validation of DQ assessment techniques used in a process. Such validation is carried out starting from the analysis of the output results of a process, the input data of which are appropriately altered in order to simulate critical situations that can really occur within processes. This analysis provides: (i) an objective ranking of the DQ dimensions to consider for the quality-aware redesign of the considered business process; (ii) the detection of abnormal situations in the process execution; and (iii) an evaluation of the effectiveness of the quality controls used in the organizations. In order to better describe the approach and to show its effectiveness, we presented how we used the methodology for validating DQCs implemented in processes related to the risk management area of a financial institution. The given approach has been tested in the financial case; however, the method is generally applicable for all processes that result in an evaluation at the end of the process and for which the quality of the result depends on this evaluation.

Q4

Based on what is shown from a theoretical perspective and on the results achieved in the analyzed case study, possible ideas for future work include the extension of the methodology to include other quality dimensions, in particular, focusing on timeliness and its evaluation. Further analysis is also needed on the criteria for inserting faults in a systematic way, providing an appropriate tool for a systematic analysis of the identified types of faults relevant for the process.

ACKNOWLEDGMENTS

The authors would like to thank the financial institution that provided specifications and data for the case study in the application scenario.

REFERENCES

Carlo Batini and Monica Scannapieco. 2016. *Data and Information Quality: Dimensions, Principles and Techniques*. Springer.

Eta S. Berner, Ramkumar K. Kasiraman, Feliciano B. Yu, Midge N. Ray, and Thomas K. Houston. 2005. Data quality in the outpatient setting: Impact on clinical decision support systems. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'05)*.

BSI. 2015. Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Measurement of Data Quality. ISO/IEC 25024:2015. Retrieved December 2, 2017, from https://www.iso.org/standard/35749.html.

Fabrizio De Amicis and Carlo Batini. 2004. A methodology for data quality assessment on financial data. *Studies in Communication Sciences* 4, 2, 115–137.

J. Debattista, S. Auer, and C. Lange. 2016. Luzzu—a framework for linked data quality assessment. In *Proceedings of the 2016 IEEE 10th International Conference on Semantic Computing (ICSC'16)*. 124–131. DOI: http://dx.doi.org/10.1109/ICSC.2016.48

Yotam Evron, Pnina Soffer, and Anna Zamansky. 2017. Incorporating data inaccuracy considerations in process models. In *Enterprise, Business-Process and Information Systems Modeling*. Lecture Notes in Business Information Processing. Springer, 305–318.

Clarissa Falge, Boris Otto, and Hubert Österle. 2012. Data quality requirements of collaborative business processes. In *Proceedings of the 45th Hawaii International Conference on Systems Science (HICSS'12)*. IEEE, Los Alamitos, CA, 4316–4325. DOI: http://dx.doi.org/10.1109/HICSS.2012.8

Maria Grazia Fugini, Barbara Pernici, and Filippo Ramoni. 2009. Quality analysis of composed services through fault injection. *Information Systems Frontiers* 11, 3, 227–239.

Paul Glowalla and Ali Sunyaev. 2014. Process-driven data quality management: A critical review on the application of process modeling languages. *Journal of Data and Information Quality* 5, 1–2, 7:1–7:30. DOI: http://dx.doi.org/10.1145/2629568

Cecil Hastings, Frederick Mosteller, John W. Tukey, and Charles P. Winsor. 1947. Low moments for small samples: A comparative study of order statistics. *Annals of Mathematical Statistics* 18, 3, 413–426. DOI: http://dx.doi.org/10.1214/aoms/1177730388

Geir Inge Hausvik. 2017. The role of information quality in healthcare organizations: A multi-disciplinary literature review. In *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS'17)*.

Benjamin T. Hazen, Christopher A. Boone, Jeremy D. Ezell, and L. Allison Jones-Farmer. 2014. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics* 154, 72–80. DOI: http://dx.doi.org/10.1016/j.ijpe.2014.04.018

Mitra Heravizadeh, Jan Mendling, and Michael Rosemann. 2009. Dimensions of business processes quality (QoBP). In *Business Process Management Workshops*. Lecture Notes in Business Information Processing. Springer, 80–91. DOI: http://dx.doi.org/10.1007/978-3-642-00328-8_8

Mei-Chen Hsueh, Timothy K. Tsai, and Ravishankar K. Iyer. 1997. Fault injection techniques and tools. *Computer* 30, 4, 75–82.

Kyriakos Kritikos, Barbara Pernici, Pierluigi Plebani, Cinzia Cappiello, Marco Comuzzi, Salima Benbernou, Ivona Brandic, Attila Kertész, Michael Parkin, and Manuel Carro. 2013. A survey on service quality description. *ACM Computing Surveys* 46, 1, 1. DOI: http://dx.doi.org/10.1145/2522968.2522969

Yang W. Lee, Diane M. Strong, Beverly K. Kahn, and Richard Y. Wang. 2002. AIMQ: A methodology for information quality assessment. *Information and Management* 40, 2, 133–146. DOI: http://dx.doi.org/10.1016/S0378-7206(02)00043-5

Bina Lehmann. 2003. Is it worth the while? The relevance of qualitative information in credit rating. In *Proceedings of the EFMA 2003 Helinski Meetings*.

Hema S. Meda, Anup K. Sen, and Amitava Bagchi. 2010. On detecting data flow errors in workflows. *Journal of Data and Information Quality* 2, 1, 4:1–4:31.

Helen-Tadesse Moges, Karel Dejaeger, Wilfried Lemahieu, and Bart Baesens. 2012. A total data quality management for credit risk: New insights and challenges. *International Journal of Information Quality* 3, 1, 1–27. DOI: http://dx.doi.org/10.1504/IJIQ.2012.050036

Martin Ofner, Boris Otto, and Hubert Österle. 2012. Integrating a data quality perspective into business process management. *Business Process Management Journal* 18, 6, 1036–1067. DOI: http://dx.doi.org/10.1108/14637151211283401

Payam Hassany Shariat Panahy, Fatimah Sidi, Lilly Suriani Affendey, and Marzanah A. Jabar. 2014. The impact of data quality dimensions on business process improvement. In *Proceedings of the 2014 4th World Congress on Information and Communication Technologies (WICT'14)*. 70–73. DOI: http://dx.doi.org/10.1109/WICT.2014.7077304

Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. 2002. Data quality assessment. *Communications of the ACM* 45, 4, 211–218.

Thomas C. Redman. 1996. *Data Quality for the Information Age*. Artech House.

Thomas C. Redman. 1998. The impact of poor data quality on the typical enterprise. *Communications of the ACM* 41, 2, 79–82.

Thomas L. Saaty. 2001. *Fundamentals of the Analytic Hierarchy Process*. Springer, Netherlands, 15–35. DOI: http://dx.doi.org/10.1007/978-94-015-9799-9_2

Monica Scannapieco, Barbara Pernici, and Elizabeth M. Pierce. 2002. IP-UML: Towards a methodology for quality improvement based on the IP-MAP framework. In *Proceedings of the 7th International Conference on Information Quality (IQ'02)*. 279–291.

Rainer Schmidt, Wided Guédria, Ilia Bider, and Sérgio Guerreiro (Eds.). 2016. *Enterprise, Business-Process and Information Systems Modeling: 17th International Conference, BPMDS 2016, 21st International Conference, EMMSAD 2016, Held at*

CAiSE 2016, Ljubljana, Slovenia, June 13-14, 2016, Proceedings. Lecture Notes in Business Information Processing, Vol. 248. Springer. DOI: http://dx.doi.org/10.1007/978-3-319-39429-9

Ganesan Shankaranarayanan, Richard Y. Wang, and Mostapha Ziad. 2000. IP-MAP: Representing the manufacture of an information product. In *Proceedings of the International Conference on Information Quality (IQ'00).* 1–16.

Pnina Soffer. 2010. *Mirror, Mirror on the Wall, Can I Count on You at All? Exploring Data Inaccuracy in Business Processes.* Lecture Notes in Business Information Processing, Vol. 50. Springer, 14–25. DOI: http://dx.doi.org/10.1007/978-3-642-13051-9_2

Diane M. Strong, Yang W. Lee, and Richard Y. Wang. 1997a. 10 potholes in the road to information quality. *Computer* 30, 8, 38–46.

Diane M. Strong, Yang W. Lee, and Richard Y. Wang. 1997b. Data quality in context. *Communications of the ACM* 40, 5, 103–110. DOI: http://dx.doi.org/10.1145/253769.253804

Richard Y. Wang and Diane M. Strong. 1996. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems* 12, 4, 5–33.

Mathias Weske. 2012. *Business Process Management: Concepts, Languages, Architectures*. Springer.

Haissam Ziade, Rafic A. Ayoubi, and Raoul Velazco. 2004. A survey on fault injection techniques. *International Arab Journal of Information Technology* 1, 2, 171–186.

Received February 2016; revised August 2017; accepted September 2017

Author Queries

Q1: AU: OK as edited?

Q2: AU: Please provide complete current mailing and email addresses for all authors.

Q3: AU: OK as edited?

Q4: AU: OK as edited?