Deploying machine learning based data quality controls – Design principles and insights from the field

Valerianne Walter[1], Andreas Gyoery[1], Christine Legner[1]

[1] Faculty of Business and Economics (HEC), University of Lausanne, Switzerland
{valerianne.walter,andreas.gyoery,christine.legner}@unil.ch

Abstract. Machine Learning (ML) has become one of the most promising technological advances for enterprises to improve manual, highly resource- and time-consuming processes. Developing and deploying these ML based systems in an organizational setting, however, is linked to a range of processual and technical requirements and implications that researchers and enterprises have only started to comprehend. Based on an Action Design Research approach, this study develops a ML based solution for data quality (DQ) controls, an essential instrument in Data Quality Management. We synthesize our findings through a set of design principles for ML based DQ controls that describe key components in the three phases from proof-of-concept to deployment and business process integration. Our findings lay groundwork for future research in the field of ML based systems for DQ and contribute to the broader IS discourse on how to embed learning-based systems in real-world organizational contexts.

Keywords: *data quality, data quality controls, rule-based systems, ML systems*

1 Motivation

Recent technological breakthroughs have opened the way for a new generation of systems that promise to fundamentally change the tasks that can be automated within an enterprise [1–3]. A field where enterprises are seeking to adopt smarter systems is Data Quality Management (DQM). Despite the business-criticality of data and almost 30 years of active research, DQM remains a challenging topic for organizations [4, 5]. As of today, achieving data quality in day-to-day business is still linked to highly manual, highly resource- and time-consuming processes that limit the amount of data for which high quality can be assured [5, 6]. By developing systems that continuously learn from patterns in the data, researchers and enterprises hope to facilitate the detection of errors, duplicated data, or completion of missing data. Approaches based on machine learning (ML) for DQM have therefore gained growing attention in the research community, yielding a large body of algorithms, techniques, and tools [7–11]. To date, existing studies focus either on developing ML techniques/systems for individual data curation tasks [7–11] or identifying potential application areas along the data value chain [12]. Despite the potential that ML holds for DQM, however, we still ignore how ML based DQ techniques can be fully integrated into the organization's

17th International Conference on Wirtschaftsinformatik,
February 2022, Nürnberg, Germany

existing processes and IT landscape. We further lack an understanding how the emerging learning-based approach differs from traditional, rule-based DQ approaches.

In a first attempt to close this gap, we conduct an Action Design Research (ADR) study to investigate DQ controls, a key instrument in DQM. These controls verify periodically that data is consistent with business goals and business rules in place. As observed in the context of a large German fashion and retail company, controls are typically based on checks hard-coded in IT systems, an effort difficult to sustain for more complex and wide-ranging controls. With this study, we aim to understand how ML based DQ controls can be integrated and deployed in an organization's process and IT landscape. We thereby answer the following two research questions:

- *RQ1: What are design principles for fully integrated ML based DQ controls?*
- *RQ2: How does the learning-based approach differ from traditional, rule-based approaches?*

By proposing design principles for ML based DQ controls and by identifying differences to traditional DQ approaches, we lay groundwork for future research in the field of ML based systems for DQ. We further contribute to the broader discourse around the productive deployment and use of ML systems in enterprises [13, 14] by providing empirical evidence of a fully integrated ML system. The remainder of the paper is structured as follows. First, we review DQ and DQM concepts and the gaps that motivate our research. We then introduce ADR as our research method. The subsequent parts present our findings along the ADR phases. We synthesize our findings as design principles and compare ML based to traditional DQ controls. We conclude with a discussion as well as outlooks on future research directions.

2 Background

2.1 Data quality and data quality management

Although DQ is considered critical, its definition is not trivial. The most popular understanding is that data is of high quality if it can satisfy the requirements for its intended use in a specific situation, a concept referred to as *fitness for use* [15, 16]. DQ can thereby be conceptualized as multi-dimensional construct consisting of a set of quality attributes, called DQ dimensions [16]. Inspired by quality management systems for production processes, researchers at MIT developed the concept of *Total Data Quality Management* [17], laying the ground for modern DQ methodologies. [18] identified three steps common to all DQ methodologies. The first step *State reconstruction* aims at establishing a baseline of information on the context of the data in scope (e.g., processes, systems, services). Based on the data collected, the second step *Assessment/measurement* defines the requirements towards the data. These requirements are translated into suitable DQ metrics used to measure the data and understand the current level of DQ. Following the identification and quantification of data defects, the third and final step *Improvement* seeks to identify and implement (reactive or proactive) DQ measures to improve the quality of the data. In a typical enterprise, each of these steps involve significant manual work of domain experts, a resource typically expensive and scarce within an organization [18, 19].

2.2 Data quality controls

Continuously controlling the quality of data is key to a sustainable DQ strategy [18, 20, 21]. To do so, DQ controls need to be set up to enforce business rules that in turn ensure DQ in one or more DQ dimension relevant to the data consumer. These DQ controls consist of manual or automatic checks and procedures in the data production processes that validate data values or conversely identify data defects [18, 20, 21]. They can be divided along the types of DQ dimension they enforce [20, 22] (see Table 1 for examples**)**. Beyond the checks, a DQ control must include the organizational follow-up processes to monitor the results of the check, through dashboards or control charts, as well as remediation measures for defects detected [18, 20, 21].

Table 1. Examples of DQ controls adapted from [23]

Dimensions	**Type of DQ controls**
Accuracy	Correctness of the data
Completeness	Completeness of data regarding e.g., schema, column, or population
Consistency	Consistency between instances or fields of the data
Timeliness	Checks that data is up to date with respect to the task

The establishment of larger DQ control frameworks that encompass several controls has gained particular traction in the field of master data management, resulting in methodological, organizational, processual, and technical approaches for establishing such frameworks (e.g., [4, 20, 22]). Traditionally, the actual implementation of controls in enterprises relies on non-learning approaches and therefore requires the implementation of a set of fixed rules [4, 22]. Not only is the development of the checks and their implementation extremely resource intensive, but the system also requires continuous update and maintenance to cope with changing rules and DQ requirements from users. The larger the scope of the controls, the higher the resources bound by the process, setting limits to the level of DQ that can be achieved.

2.3 Data quality tools and ML for data quality

DQ tools to support and automate DQ tasks have been a focus point of database and web research [24] and build on non-learning and learning approaches [25, 26]. Notable categories of tools include data profiling [27] and data preparation [28]. As mentioned in section 2.2, non-learning-based approaches rely on user-provided rules and are based on fixed constraints specified by a user or external references. Learning-based approaches leverage statistical learning methods to continuously learn to detect and clean errors based on patterns in the data [25]. The learning-based approaches have gained increasing traction in DQM research [12, 26, 29]. Based on techniques and tools from research and practice, [12] have identified nine usage scenarios providing a first indication of potential DQ routines that can be automated with ML. As DQ systems can only work in interplay with domain experts, researchers are seeking to develop systems that build on *human-in-the-loop ML* approaches [30, 31]. Notable examples of systems include *TAMR* [32], for data deduplication and integration, *Magellan* [33], for entity

resolution, *HoloClean* [34], for data cleaning. These systems are considered state-of-the-art for their respective data curation task [26]. As observed by [29], however, DQ systems developed and discussed in research largely focus on developing isolated systems for specific problems, but ignore the enterprise's wider context. In a first attempt to close the gap, [29] have developed a ML based DQ solution for data cleaning, more suited to the needs of DQ managers. As a prototype, the tool, however, does not account for the challenges of productive deployment into the existing processes and infrastructural realities of an organization. We therefore lack a comprehensive understanding of how to design fully integrated ML DQ systems that account for the processual and technical elements to be in place, for these systems to work.

3 Research design: Action Design Research

To address this gap and answer our two research questions, we opt for Action Design Research (ADR). ADR is specifically designed to tackle the problem in a particular organizational setting and develop an ensemble artifact "that addresses the class of problems typified by the encountered situation" [35]. The project described here is part of a research collaboration on learning-based DQM approaches with a major German fashion and retail company, called FashionCo in the following. FashionCo is a suitable candidate for this type of research for the following reasons: (1) their high maturity of the overall DQM frameworks and systems, (2) the willingness of the DQM team to explore new and more scalable approaches to DQM and provide researchers access to internal experts, data, and systems, (3) the tangible pain points FashionCo faces in the product data creation process that are hard to solve with traditional methods. The use case presented is a DQ use case, where we designed and implemented ML based controls, from proof-of-concept to productive use.

In line with ADR guidelines, the design and implementation consisted of four stages, conducted from June 2020 to August 2021 (see Table 2 for a summary of these phases, the actors involved, and research activities). The first step *Problem formulation* aimed at understanding the organizational setting and challenges the company and the DQM team were facing with the implementation of traditional DQ controls. *Building, Intervention, Evaluation (BIE)* consisted in the development and evaluation of said ML based DQ controls. In analogy to the ML development phased proposed by [13], we ran three BIE cycles: *Development of the ML model* and *Deployment of the ML model* as well as *Integration into business process.* During these phases, the main researcher and first author worked closely with relevant actors from FashionCo, which included: the DQM team, as specialists for internal DQM methods, processes, and systems and eventual owners of the solution developed; the finance team, as the business experts and data users, who detect the data defects; product managers, as data creators who have to correct the defects at source and would use the ML system's results. We further involved IT experts to support the implementation of automation procedures of ML results in FashionCo's application landscape. To involve these actors, the first author conducted weekly meetings with the DQM team. She conducted additional working sessions with relevant actors to clarify requirements, share and evaluate the results and

the design. The research activities were documented in meeting minutes, thought protocols, and a comprehensive project documentation. *Reflection and learning* and *Formalization of learnings* involved the review of the project results in the extended research team and their discussion in the light of academic literature. In addition, semi-structured interviews were conducted with three independent consultants with expertise in building similar ML based DQ systems for comparable organizations. These interviews consisted in a walk-through of the current design and in reflection on similarities and differences to implementations they have built in the past. The design principles were reviewed by two DQM researchers familiar with ML setups.

Table 2. ADR stages, actors, and activities

Stage	Actors involved	Activities
Problem formulation Understand business problem, data, processes, and systems (06/2020)	Researcher, DQM team lead, finance team (team lead, team members)	• Two meetings with finance team to clarify scope and current requirements (06/2020) • Exploration of data • Weekly 1h exchange with DQM team lead on progress
BIE cycle 1 - Development of ML based DQ control (07/2020-09/2020)	Researcher, DQM team lead, finance team, product managers (80 persons contacted via e-mail)	• Collection and preparation of data and testing of different ML algorithms • Weekly 1h exchange with DQM team lead on progress and intermediate results • Evaluation of results with finance team by review of results (08/2020) • E-mails sent to product managers to initiate corrections (09/2020)
BIE cycle 2 - Deployment of ML based DQ control (12/2020-05/2021)	Researcher, DQM team, big data platform engineer	• Adaption of code base to internal infrastructure and productive use • Weekly 1h exchange with DQM team lead on progress and intermediate results • Evaluation of design with DQM team (10/06/2021)
BIE cycle 3 - Integration into DQ monitoring and correction process (05/2021-06/2021)	Researcher, dashboarding team, message broker engineer, 2 product managers	• Technical and organizational integration into control process • Weekly 1h exchange with DQM team lead on progress • Evaluation of design with product managers (06/2021)
Formalization of learnings Reflection of solution (06/2021-08/2021)	Research team, DQM team lead, Consultant 1, 2 (associate and solution architect), Consultant 3 (director of data science)	• 1h interview consultant 1, 2 (29/06/2021) • 1h interview consultant 3 (11/08/2021) • Review of design principles by two DQM researchers (26/08/2021)

4 Introducing ML based DQ controls at FashionCo

4.1 Problem formulation

Context. As global fashion and retail company, FashionCo faces the challenge of a fast-changing seasonal product portfolio with around 100,000 active products and several 10,000 new products per season. The fast-paced product creation process leaves limited time and resources for product managers to create the respective master data, leading to a high risk of missing or incorrect information. To improve the product data creation process, FashionCo's DQM team has developed a centralized DQ framework to run validations for product master data, that they keep gradually improving.

DQ control framework. The DQM's team approach to expand the DQ control framework comprises five elements. Starting from a business use case, the DQM team first elaborates the business rules with business experts. It then implements these as hard-coded checks in the internal DQ platform (*creation of control*). The checks are executed daily in an automated manner (*run of control*). Due to the efforts required to elicit, implement, and maintain the checks, controls are limited to consistency and completeness checks, spanning over 1-3 attributes (*scope of control*). As mitigation measure, detected data defects are returned to the initial creators of the data, the product managers, via a dashboard and specific correction lists per product manager. The product managers then correct the entries in the source system to avoid later fixes in downstream systems (*follow-up of control*). In case the business rules change, the checks are updated manually by the DQM team (*update of control*).

Use case. The use case in scope focuses on an attribute containing information about business partners which is of critical importance for financial planning and correct payments. Due to lack of guidance and training, product managers are not always aware of the criticality of the attribute, which often leads to incomplete or inaccurate values. To avoid these data defects, the finance team established a manual review process and corrected the entries in the financial system (not in the source system for product data). This resulted in an effort of at least 10 full man days per season. The finance team checked around 200 active business partners and 50,000 articles every season by verifying two other attributes that contained indications on the correct value for the attribute. To help the finance team and introduce automatic checks in the centralized DQ control framework, the DQM team had already tested the implementation of hard-coded checks for a limited amount of business partners. While faster than the manual checks performed by the finance team, one member of the finance team stated that "we do not necessarily know the rules, we check each entry one by one and check if there is a reference that we recognize" [finance team lead], further adding "it would be a lot of work to give you all the rules, nearly impossible". Furthermore, each season brings new business partners with a new set of rules requiring constant update of the system. The complexity of the rules and their changing nature made FashionCo look for a solution that could automate and accelerate the data validation and correction.

4.2 Designing and implementing ML based DQ controls at FashionCo

BIE cycle 1: Development of ML based DQ control

Based on the problem analysis, the researcher and DQM team chose to test a learning-based approach and develop a ML model that predicts the potential value of the attribute based on patterns in historical product data. Activities performed in this cycle can be structured along the components *Data*, *Preprocessing*, and the *ML algorithm*.

Data. FashionCo's product data consists of around 200 attributes, of which only few are relevant for the use case. To avoid overtraining, a careful selection of attributes that contain indications of the correct business partner was required. Next to the two initial attributes used by the finance team, five more attributes were identified through basic correlation analysis, yielding seven relevant attributes, four categorical attributes for product categories and three free text attributes for product description. Acquiring historic data reflecting the business rules in place was a major challenge, as data was corrected in downstream systems and not updated at source. A significant amount of time was therefore spent gathering, consolidating, and correcting data from downstream systems to create an initial, quality-assured training data set.

Preprocessing. To apply the ML algorithm, the raw data needed to be preprocessed. Quality of the three text attributes varies significantly as the descriptions are not standardized and the text may contain different types of typographical errors. The text is therefore normalized (incl. lowering, removal of extra whitespaces, tokenizing to 2-n-grams) and the attributes are one-hot-encoded.

ML Algorithm. From an algorithmic perspective, the problem can be broken down to a classification problem. Based on previous historical data and attributes, the algorithm learns the patterns to predict the value of a specific attribute. Once it has learned the patterns, the model can be used on a "dirty" dataset and predict the correct value. Several ML classification algorithms (decision trees, random forest, support vector machines, logistic regression, neural networks with different hyperparameters) were tested. Initial tests showed varying performance with an F1-score of 83%-95% on average. However, cross-validation revealed that overtraining occurred in almost all cases requiring a further careful pre-selection of features. Early exchange with finance managers further showed that users often needed to know why the ML model detected an error, especially if the ML model was wrong. We therefore opted for an ensemble of linear SVM approach that allow to deduce the attributes and values that led to the decision through the weights of the support vectors. It also makes it easier to surveil if the model has overtrained on irrelevant features. The final algorithm yielded an average F1-score of 92%, which was deemed adequate by the DQM team. When reviewing the results, one finance manager stated that "it's impressive what the machine can do", "we wouldn't have found the business partner x because we do not check that field usually, it would be too much to review really on top of everything else" or "I wasn't aware that the abbreviation x meant business partner y, that's good to know". In its initial run, the model found 780 valid errors vs. 512 in the manual check and around 15 false positives. The finance managers affectionately named the model "The Machine".

BIE cycle 2: Deployment of ML based DQ control

With a first model in place, it was decided to deploy the model to production. This step can be further divided into the components *Running*, *Retraining*, and *Monitoring*.

Running. After local prototyping of the ML model with hard-copied data extracts, the first step was to identify an adequate infrastructure for deployment, to run the model on a continuous basis ("We cannot run this on a local machine forever" [DQM team lead]). The existing DQ platform did not provide the capability to properly run custom code. Therefore, we decided to deploy the model on FashionCo's big data platform, built on Amazon Web Services, that includes a big data lake mirroring the data of the company's main systems and a continuous integration and deployment environment. This platform provides a stable, daily updated supply of the data to control, the possibility to automate pre-processing steps and model execution. As the big data platform is placed further downstream in the data supply chain, data has already passed several standardization steps at this stage. The change in stack and data required major adjustments to adapt the model to the new data and to automate daily execution of checks.

Retraining. As the rules around the attribute in scope evolve every season, the initial model got quickly outdated. We thus needed to implement a retraining process in line with the data creation and correction process. Product data is created in milestones while the correction process takes place simultaneously. At the peak of the product creation process, new products are entered and corrected daily at high frequency. Training on the previous, quality-assured seasons' data runs once per season, retraining on the current season runs daily to capture any changes in the data.

Monitoring. Closely linked to the retraining process, is the monitoring process. We needed to provide the DQ manager with the possibility to surveil retraining and the overall performance of the model in case unexpected errors in the re-training occurs. Through a dedicated user interface, the DQ manager can monitor the model evaluation scores (F1-score, precision, recall), the current number of open corrections and additional information, on which features the decisions are made. The user interface also provides the possibility to remove features that are considered irrelevant by the DQM manager. This gives the model further context beyond the statistical relations within the data. The final design and implementation of the ML based DQ controls were discussed with two members of the DQM team in a walk-through of the components. While they understood that the different components were necessary, they found the implementation of an ML system also to be "very complex" compared to their traditional approach. Noting that many of the components may be re-used (the CI/CD pipelines, the algorithm), they still wondered "if it's that complicated, will we be able to implement more controls?" [DQM team member].

BIE cycle 3: Integration into DQ monitoring and correction process

The last phase focused on the *Technical integration* and the *Organizational integration* of the ML based DQ control. To establish one single channel for corrections, it was decided to integrate the results into the existing DQ dashboard and into the existing DQ monitoring and correction process ("We cannot possibly have five communication channels to the product managers, it will drive them crazy" [DQM team lead]).

Technical integration. For integrating the results of the ML based DQ control into the current DQ dashboard, messages are sent through the internal message broker system and are forwarded to the internal DQ dashboard, where they are shown along other DQ checks. The messages generated by the ML based DQ control state the product affected along with feedback on the detected errors ("*Check if xxx should be added, attribute yyy contains/is zzz*") and a confidence score. To avoid too many false positives and increase user acceptance, the DQM team lead suggested that findings with a confidence score below 90% were to be excluded from the checks.

Organizational integration. Finally, we informed the product managers about the new DQ control and explained them how to use the results in the DQ monitoring and correction process. Special emphasis was put on the uncertainty of results and the implications of the confidence score. Feedback during user acceptance test was positive ("It definitely makes sense to check this attribute", "Yes, please integrate this into the dashboard" [product manager]), but they found the *confidence score* "confusing". It was decided to remove the field from the dashboard going forward.

The control has now run for one season, discovering around 700 data defects. Since the control was integrated into the existing correction process, the new control caused little disruption on product manager side. In informal discussions for the wrap-up of the project, one finance manager stated "the field had such a bad data quality before, now it's much better thanks to the machine. We can run our reports in the reporting systems directly rather than manually fixing them by hand first" [finance team lead]. Another stated that "it saves 80% of the time I spent on data validation" [finance team member]. While the use case only covered one attribute, a DQM manager noted that "it took some time, but this is an approach we could easily reuse on other attributes as well", thus being "a potential blueprint for further projects".

5 Findings

5.1 Design principles of integrated ML based DQ controls

Based on the insights gained in this ADR study, we were able to derive design principles for ML based DQ controls along the three phases from proof-of-concept to deployment and business process integration. Table 3 summarizes the eight design principles that relate to the following key components: *Data, Preprocessing, Algorithm* represent the technical core of the control (as reflected in e.g., [29]). *Running, Retraining, Monitoring* address the organizational and technical elements needed for the model to work in an organizational setting. This includes not only the infrastructure to run the model, but also an adequate data supply, a retraining strategy that ensures that the model stays up to date, as well as a monitoring strategy that allows the DQM manager to intervene. The two last elements *Technical integration* and *Organizational integration* describe how the results are integrated into the organization's DQ monitoring and correction process, both from application and processual perspective.

Table 3. Design principles for fully integrated ML based DQ controls

Development of ML model	
1.1. Data	Training dataset to learn DQ rules that includes: • Reference points from which implicit business rules can be derived • Historical data that follows expected, valid, implicit business rules • Enough instances in the data to learn the different business rules from
1.2. Pre-processing	Processing steps adapted to the training dataset that include: • Normalization and vectorization of categorical and textual attributes • Careful selection of attributes and features
1.3. Algorithm	An ML algorithm that can learn the implicit business rules and that allows for: • Quality of predicted correction adequate to the task at hand • Control of features taken for prediction of a data defect • Explanation of potential data defect
Deployment of ML model	
2.1. Running	Integration into the existing data supply chain, based on an infrastructure that: • Is scalable, allows to run custom code and to build automation pipelines • Has a reliable supply of the data to control and position in data supply chain adapted to the frequency of changes of the data production process and the frequency of follow-up process
2.2. Retraining	Retraining strategy that foresees: • An automated, regular retraining procedure to capture new business rules • A careful selection of adequate data instances to be added to the *Data* component on which to base retraining • Is adapted to the frequency of changes of business rules in the data production process controlled
2.3. Monitoring	Monitoring strategy that foresees: • A DQM manager to monitor performance of the model • Regular monitoring and review process of the model • User interface to monitor and help the model
Integration into business process	
3.1. Technical integration	Integration of results of into user systems with: • A seamless integration into the user interface • Feedback indicating error, suggestion for the correct value, and explanation
3.2. Org. integration	Integration of control results into the organization that builds on: • Users that are aware of uncertainty related to the results and possibility for false positives and can therefore act on the corrections proposed

The discussion and review of the design principles with experts from academia and practice yielded three interesting observations that give more depth to the findings.

Data. Finding/producing an initial dataset with business rules, that the algorithm can learn from, remains one of the most difficult and time-consuming tasks when setting up ML based DQ controls (as in many other data science/ML projects [13, 36]). An expert noted that it makes sense to explore how ML can support at these earlier stages.

Transparency. Transparency eases human-machine interactions, but experts noted that the level of transparency achieved in this use case and the feedback provided is rather unusual. With more complex business rules, transparency is rarely feasible requiring more thorough onboarding of users. Indeed, transparency remains one the most difficult and researched challenges in deployment of ML systems [13, 37, 38].

Maturity of other organizations. While agreeing with the components themselves, one expert stated that few companies get past the development phase, as they "oftentimes lack that type of process orientation" and/or "general readiness and maturity of infrastructure and frameworks" [consultant 1], a challenge reflected in literature and general ML projects [13].

5.2 ML based DQ controls vs. traditional approaches

To analyze the *differences between the ML based DQ controls and traditional, non-learning approach* we refer to FashionCo's DQ control framework (see section 4.1) and summarize our findings in Table 4. In terms of *scope*, we observe that traditional DQ controls are limited to well-defined completeness and consistency checks, whereas ML based controls allow to include "fuzzy" business rules. Thereby, they extend DQ control's coverage to include consistency and accuracy checks and address more complex DQ dimensions. The ML based approach introduces a fundamental shift in the *creation of the control* – from "learning business rules from experts, validated with data" to "learning business rules from data, validated with experts". A further shift can be observed for the *update*, as in the learning-based approach the control gets updated through retraining the ML model. End-users further expect a seamless integration into the *Follow-up process* no matter the approach.

Table 4. Non-learning DQ approach vs. Learning DQ approach

	Non-learning DQ approach	Learning DQ approach
Scope	• Precise, well-defined business rules • Limited coverage, focusing on few DQ dimensions (completeness and consistency)	• More variation and "fuzziness" of business rules • Higher coverage, including additional DQ dimensions (consistency and accuracy)
Creation	• Business rules learned from business experts, validated with data • Hard-coded in DQ platform by DQM manager	• Business rules are learned from data, validated by business experts • Custom development by data scientist built on big data infrastructure
Run	• Automatic checks performed against constraints on regular basis	• Automatic checks performed against the model on regular basis
Update	• Punctual update of checks to add/remove/change procedure in place • Manual by DQM manager	• Retraining of model based on updated data and monitoring of models and its results • Little to no intervention
Follow-up process	• Same channel • Similar follow-up process • Accepted as such by users	• Same channel • Similar follow-up process • Training to raise awareness of users for uncertainty of results/false positives

Discussion with experts yielded more observations that complement these findings.

Implicit versus explicit. As noted by one of the experts, "the very nature of the control changes" [consultant 1], as business rules are not elicited explicitly anymore.

ML allows for "checks that were not expected as such or that business did not think of phrasing explicitly" [consultant 1]. Nonetheless, the training data needs to follow implicit business rules, so that the model can discover these implicit rules and then check if they are followed. This allows to check against more rules with more complexity but is also prone to over-training, therefore requiring a careful up-front selection of features.

Technical complexity. From a process and architectural perspective, implementing ML based DQ controls does not differ much from other ML projects. However, experts emphasize that the approach is new to the enterprise data management discipline and requires an entirely different set of skills and technology than is generally found in data management. Overall, the complexity of implementing a ML based DQ control was by far greater than implementing fixed rules. This limits the use cases viable for ML based DQ controls and the organizations that may be able to afford to implement these.

Human involvement. Previously, experts were needed to elicit, implement, and update DQ checks. Their involvement now shifts to organizing the data to learn from, to setting up the model, to monitoring and nudging it into right direction. Thus, a shift towards more up-front human involvement in setting up the control, less in production.

6 Conclusion

Based on a real-world DQ use case at a global fashion and retail company, our study provides rich insights into ML based DQ controls in a complex enterprise context. We identify eight design principles for integrated ML based DQ controls that describe the design of key components along the three phases from proof-of-concept to deployment and business process integration. By analyzing the changes to the prevailing DQ approach, we demonstrate that learning-based DQ approaches will significantly impact DQ practice going forward. Our findings provide a starting point for the design of end-to-end ML based DQ systems and contribute to the research stream on ML for DQ.

Our research also contributes to the broader debate in IS research around the paradigm shift from systems where logic is coded (rule-based systems) to systems that learn from data (probabilistic systems) [1, 2, 39]. We consider DQM as exemplary of clerical tasks that can be automated or augmented with ML. By outlining design principles that address the challenges and complexities of real-world use cases, our study is a first step towards a more comprehensive understanding of embedding learning-based systems in real-world organizational context.

We must, however, acknowledge several limitations in this research. While building DQ controls is not specific to the fashion industry, FashionCo's maturity in terms of DQM frameworks, processes, and applications were pre-requisites to a successful ML integration. Future research will need to more closely investigate the contextual factors and generalizability of our findings. We also see research opportunities related to the theoretical grounding of our findings, especially by incorporating the theoretical lenses of organizational routines [40, 41] and delegation to novel IS agents [1].

References

1. Baird, A., Maruping, L.: The Next Generation of Research on IS Use: A Theoretical Framework of Delegation to and from Agentic IS Artifacts. MISQ. 45, 315–341 (2021).
2. Lacity, M., Willcocks, L.P.: Robotic process and cognitive automation: the next phase. SB Publishing, Ashford (2018).
3. Maedche, A., Legner, C., Benlian, A., Berger, B., Gimpel, H., Hess, T., Hinz, O., Morana, S., Söllner, M.: AI-Based Digital Assistants: Opportunities, Threats, and Research Perspectives. Bus Inf Syst Eng. 61, 535–544 (2019).
4. Otto, B., Österle, H.: Corporate Data Quality: Voraussetzung erfolgreicher Geschäftsmodelle. Springer Gabler, Berlin, Heidelberg (2016).
5. Redman, T.C.: Data Quality Management Past, Present, and Future: Towards a Management System for Data. In: Sadiq, S. (ed.) Handbook of Data Quality. pp. 15–40 (2012).
6. Batini, C., Scannapieco, M.: Introduction to Information Quality. In: Data and Information Quality. pp. 1–19 (2016).
7. Laure, B.-E., Angela, B., Tova, M.: Machine Learning to Data Management: A Round Trip. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE). pp. 1735–1738 (2018).
8. Ilyas, I.F., Chu, X.: Trends in Cleaning Relational Data: Consistency and Deduplication. FNT in Databases. 5, 281–393 (2015).
9. Doan, A., Suganthan, G.C.P., Zhang, H., Ardalan, A., Ballard, J., Das, S., Govind, Y., Konda, P., Li, H., Mudgal, S., Paulson, E.: Human-in-the-Loop Challenges for Entity Matching: A Midterm Report. In: Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics. pp. 1–6 (2017).
10. Abedjan, Z., Chu, X., Deng, D., Fernandez, R.C., Ilyas, I.F., Ouzzani, M., Papotti, P., Stonebraker, M., Tang, N.: Detecting data errors: where are we and what needs to be done? Proc. VLDB Endow. 9, 993–1004 (2016).
11. Dong, X.L., Rekatsinas, T.: Data integration and machine learning: a natural synergy. Proc. VLDB Endow. 11, 2094–2097 (2018).
12. Fadler, M., Legner, C.: Understanding the Impact of Machine Learning on Enterprise Data Management: Taxonomic Approach. In: Proceedings of the 2019 Pre-ICIS SIGDSA Symposium (2019).
13. Baier, L., Jöhren, F., Seebacher, Stefan: Challenges in the Deployment and Operation of Machine Learning in Practice. In: Proceedings of the 27th European Conference on Information Systems (2019).
14. Hukkelberg, I., Rolland, K.: Exploring Machine Learning in a large Governmental Organization: An Information Infrastructure Perspective. In: Proceedings of the 28th European Conference on Information Systems (2020).
15. English, L.P.: Improving data warehouse and business information quality: methods for reducing costs and increasing profits. Wiley, New York (1999).
16. Strong, D.M., Lee, Y.W., Wang, R.Y.: Data quality in context. Commun. ACM. 40, 103–110 (1997).
17. Wang, R.: A Product Perspective on Total Data Quality Management. Commun. ACM. 41, 58–65 (1998).
18. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41, 1–52 (2009).
19. Parmiggiani, E., Grisot, M.: Data Curation as Governance Practice. Scandinavian Journal of Information Systems. 32, 3–38 (2020).

20. Loshin, D.: The practitioner's guide to data quality improvement. Morgan Kaufmann, Burlington, MA (2011).
21. Redman, T.C.: Getting In Front On Data. Technics Publications (2016).
22. Hüner, K.: Method for Specifying Business-oriented Data Quality Metric. University of St. Gallen, Institute of Information Management, University of St. Gallen for Business Administration, Economics, Law and Social Sciences (HSG) (2011).
23. Lee, Y.W., Pipino, L.L., Funk, J., Wang, R.: Journey to data quality. MIT Press, Cambridge (2006).
24. Sadiq, S., Yeganeh, N.K., Indulska, M.: 20 Years of Data Quality Research: Themes, Trends and Synergies. In: Proceedings of the Twenty-Second Australasian Database Conference. pp. 153–162 (2011).
25. Neutatz, F., Chen, B., Abedjan, Z., Wu, E.: From Cleaning before ML to Cleaning for ML. In: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering (2021).
26. Ilyas, I.F., Chu, X.: Data Cleaning. Association for Computing Machinery, New York (2019).
27. Abedjan, Z., Golab, L., Naumann, F.: Profiling relational data: a survey. The VLDB Journal. 24, 557–581 (2015).
28. Hameed, M., Naumann, F.: Data Preparation: A Survey of Commercial Tools. SIGMOD Rec. 49, 18–29 (2020).
29. Altendeitering, M., Guggenberger, T.: Designing Data Quality Tools: Findings from an Action Design Research Project at Boehringer Ingelheim. In: Proceedings of the 29th European Conference on Information Systems. p. 17 (2021).
30. Fails, J.A., Olsen, D.R.: Interactive machine learning. In: Proceedings of the 8th international conference on Intelligent user interfaces. pp. 39–45 (2003).
31. Krishnan, S., Wang, J., Wu, E., Franklin, M.J., Goldberg, K.: ActiveClean: interactive data cleaning for statistical modeling. Proc. VLDB Endow. 9, 948–959 (2016).
32. Stonebraker, M., Bruckner, D., Ilyas, I.F., Beskales, G., Cherniack, M., Zdonik, S., Pagan, A., Xu, S.: Data Curation at Scale: The Data Tamer System. In: 6th Biennial Conference on Innovative Data Systems Research (2013).
33. Konda, P., Das, S., Suganthan G. C., P., Doan, A., Ardalan, A., Ballard, J.R., Li, H., Panahi, F., Zhang, H., Naughton, J., Prasad, S., Krishnan, G., Deep, R., Raghavendra, V.: Magellan: toward building entity matching management systems. Proc. VLDB Endow. 9, 1197–1208 (2016).
34. Rekatsinas, T., Chu, X., Ilyas, I.F., Ré, C.: HoloClean: Holistic Data Repairs with Probabilistic Inference. Proc. VLDB Endow. 10, 1190–1201 (2017).
35. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action Design Research. MISQ. 35, 37–56 (2011).
36. Breck, E., Polyzotis, N., Roy, S., Whang, S.E., Zinkevich, M.: Data Validation for Machine Learning. In: Proceedings of the 2nd SysML Conference (2019).
37. Wanner, J., Herm, L.-V., Janiesch, C.: How much ist the black box? The value of explainability in machine learning models. In: ECIS 2020 Research-in-Progress Papers (2020).
38. Sultana, T., Nemati, H.: Impact of Explainable AI and Task Complexity on Human-Machine Symbiosis. In: AMCIS 2021 Proceedings (2021).
39. Zhang, Z., Nandhakumar, J., Hummel, J., Waardenburg, L.: Addressing the Key Challenges of Developing Machine Learning AI Systems for Knowledge-Intensive Work. MIS Quarterly Executive. 19, Article 5 (2020).

40. Pentland, B.T., Feldman, M.S.: Designing routines: On the folly of designing artifacts, while hoping for patterns of action. Information and Organization. 18, 235–250 (2008).
41. Pentland, B.T., Feldman, M.S.: Organizational Routines as a Unit of Analysis. Industrial and Corporate Change. 14, 793–815 (2005).