LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT
School of Engineering Science
Software Engineering

Elizaveta Tereshchenko

DEVELOPMENT AND IMPLEMENTATION OF THE DATA QUALITY ASSURANCE SUBSYSTEM FOR THE MDM

Examiners: Associate Professor Annika Wolff
Professor, Dr Sci. (Economics) Igor Ilin

ABSTRACT

Lappeenranta-Lahti University of Technology
School of Engineering Science
Software Engineering
Elizaveta Tereshchenko

Development and implementation of the data quality assurance subsystem for the MDM platform

Master's Thesis 2021

88 pages, 15 figures, 14 tables
Examiners: Associate Professor Annika Wolff
Professor, Dr Sci. (Economics) Igor Ilin
Keywords: master data management, master data management systems, data quality, master data, information management

Data is the fuel for artificial intelligence systems, the raw material for analytical algorithms, and the basis for business process automation systems. If decision-makers do not have timely, relevant, and reliable information, they have no choice but to rely on their intuition. Data quality becomes a crucial aspect.

The research aims to develop and implement a data quality assurance subsystem designed to improve data quality and user interaction with data. The study was carried out based on the master data management platform used in a state-owned company, which perform state cadastral registration of real estate activities. As a research method, a single case study analysis and literature review were used. The scientific novelty of the research is the development of the subsystem, which prevent enterprises from data quality problems. Considering different aspects of data quality, this research is an excellent asset to the architectures, developers, and business analysts to develop and adopt data quality assurance subsystems with master data management systems. Moreover, the methodology can also be applied to any implemented system.

ACKNOWLEDGEMENTS

I am delighted that I had an opportunity to do a Master's degree in the School of Software Engineering Science at Lappeenranta University of Technology. This work was performed with supports from many people. I am not able to refer names of everyone, but I sincerely appreciated all your invaluable support. My deepest gratitude goes to my first supervisor, Professor Annika Wolff for her invaluable support, guidance, and encouragement throughout this thesis. I want to thank my second supervisor, Dr. Igor Ilin, for sharing knowledge and resources to execute this research. Finally, I wish to express my sincere gratitude to my dearest parents, friends and the relation who always behind me every step of the way by providing unconditional support and love. Also, I am so blessed to have wonderful, caring and supportive people around me all the time. Thank you very much all of you who have been a part of my life, supporting me to reach this point. Without you all, I might not be where I am today.

Elizaveta Tereshchenko

Lappeenranta

2021

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
BPMN	Business Process Management Notation
DG	Data Governance
DQAS	Data Quality Assurance Subsystem
DWH	Data Warehouse
ESB	Enterprise Service Bus
ETL	Extract, Transform, and Load
FAR	False Acceptance Rate
FRR	False Rejection Rate
HTTP/HTTPS	Hypertext Transfer Protocol
MDM	Master Data Management
MOM	Minutes of a Meeting
OLTP	Online Transaction Processing
REST	Representational State Transfer
SIA	Service of Identification and Authentication
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
XML	eXtensible Markup Language

1 INTRODUCTION

1.1 Background

The development of information technology has made it possible to create complex information systems. Such systems can function without human intervention, have a multi-level geographically-branched structure, and are characterized by high reliability. Without high-tech information systems, it is already impossible to imagine the activities of many modern enterprises and organizations. Besides, data becomes a vital asset of the enterprise.
The creation of a single information space for modern enterprises engaged in data has become an urgent need. Against the background of the constant growth in the volume and complexity of design documentation, improving the efficiency of such enterprises is possible only if the automated systems used are integrated. In this regard, choosing a Master Data Management (MDM) system ensures the integration of the data used in the enterprise into a single information space and the management of information about the enterprise in the required volume [1]. This task is actualized even more since data from different systems are currently used in the exchange process, implying developing methods and algorithms for intersystem data exchange.

An essential component of master data management is the quality control of the data. The MDM platform should have several quality control tools, each of which is designed to solve specific tasks. Having data quality problems in an extensive enterprise system might cost the company much money. Two primary goals of MDM systems are proactively monitored and cleanse the data for all applications and keep it clean and access any data source anywhere, and deploy centralized data quality rules to improve data quality across all applications [2].

In recent years, information systems development has become the system approach, which is considered a research methodology and a modern way of managerial thinking, giving a holistic view of the organization.
To automate processes, it is necessary to synchronize input data between automated systems at all levels, solve the problem of information quality to avoid duplication and inconsistency, increase reliability, and ensure the integrity of the data.

1.2 Goals and delimitations

The goal of this research is to develop and implement the Data Quality Assurance Subsystem in the Master Data Management platform. Besides, it was studied the importance of Data Quality (DQ) in digital companies and DQ practices. A single case study analysis, based on a governmental company in Russia, was performed.

This work aims to develop and implement the Data Quality Assurance Subsystem in the Master Data Management platform. This study is designed to understand the importance of MDM systems, Data Quality (DQ) in digital companies, and existing DQ practices. A single case study analysis covers the description of the MDM platform, current advantages and disadvantages of the platform, the need for the development of DQAS, the creation of requirements for the subsystem, and the specifics of implementation stages and tools. The task is also to understand how subsystem will affect the performance and indicators of the governmental company for which the subsystem was developed and implemented. In addition, the study intends to know whether the features of the development and implementation of this case study can be generalized and applied for further use in software development and how these features influence the process of project implementation. This master dissertation is the Data Quality Assurance Subsystem (DQAS) technical design, developing the data quality assurance subsystem. It discloses the purpose and scope of its use, the characteristics of automation objects, information about the regulatory and technical documents used in the design, the description of the processes of users and personnel, the leading technology solutions for the structure of the data quality assurance subsystem, its relationships and modes of operation, the composition of functions and information, measures to prepare the automation object for putting the system into operation. The main objective of this thesis is to provide detailed and explicit information about the development and implementation of data quality assurance subsystems and, including the practices, methodology, and solutions. Moreover, the master dissertation will indicate the importance and challenges of data quality.

The main research question is, “How to develop and implement data quality assurance subsystem?”. However, this question can be refined with four more specific research questions, where the theoretical background is discussed in more detail in Section 2.

Accordingly, based on these goals, it is possible to formulate the following research questions:

RQ1. What are the existing solutions of master data management systems?
RQ2. What are the existing practices of data quality?
RQ3. How should the data quality assurance subsystem be developed and implemented?
RQ4. How effective is the development and implementation of a data quality assurance subsystem?

The answer to these research questions can help understand the data quality and develop and implement the data quality assurance subsystem.

1.3 Structure of the thesis

As part of this master dissertation, a data quality assurance subsystem was developed and implemented. The work consists of an introduction, four chapters, a conclusion and appendices. The first chapter provides an overview of modern master data management systems, provides examples of master data management systems usage, and discusses quality rules and contemporary practices. The scientific method used in this work is also justified. The second chapter is devoted to describing the existing platform, its shortcomings, the need to create a subsystem for ensuring data quality, and the justification of the methodology for developing the subsystem. The third chapter presents the data quality assurance subsystem requirements, which the project team collected and necessary for successful development and implementation. The architectural and service model is developed, taking into account the integration of the subsystem into the complex architecture of the existing platform, and the business processes for the required operation of the system are described. A data quality assurance subsystem for the master data management platform has been developed. This section is of practical importance. The fourth chapter describes the implementation methodology, provides recommendations for further implementation of systems at enterprises, and justifies the economic effect of implementing the subsystem. This chapter describes the impact and effectiveness of the development and implementation project.

The limitations of this thesis are the non-disclosure of part of the information provided by the company, so this paper does not consider the economic efficiency and payback of development and implementation.

Considering different aspects of data quality, this research is an excellent asset to the architectures, developers, and business analysts to develop and adopt data quality subsystems with enterprise systems.

2 RELATED WORK AND LITERATURE REVIEW

This chapter describes the data types and the data quality, the master data management systems. This chapter aims to explain and discuss master data management, the current trends in the management of master data, justification of the usage of data quality rules, and how the highest data validity can be achieved. A literature review of available sources was conducted to identify the research problem and clarify research questions and hypotheses. The literature review helped to understand what is known and what is unknown to determine what contribution current research will make to developing new knowledge.

In this chapter, the answers to two research questions are given:
RQ1. What are the existing solutions of master data management systems? This question is answered in Section 2.4.
RQ2. What are the existing practices of data quality? This question is answered in Section 2.6.

The first step was to define the purpose of the research literature review. The goal was to study the current trends in the management of master data and how the highest data validity can be achieved.

The next step is to determine the focus of the study. Research questions were compiled to narrow the focus of the literature review to an acceptable size and select the further direction of the study. Moreover, the criteria for including and excluding literature searches were determined [3]. The corresponding theoretical and conceptual bases for the research task were determined. Methods of data collection for the study were identified during this stage of research.

Step three was searching for the necessary papers, articles, and books in the database of university libraries and other search engines of scientific documents, such as Google Scholar, ScienceDirect, SpringerLink, IEEE Xplore DigitalLibrary4 and ResearchGate.

Search keywords were Master Data Management, Master Data Management Systems, MDM, Data Quality, Data handling, Master Data, Information management. The search was

carried out using various combinations of words described above, as well as individual words.

Finally, step four is about reading and critically evaluating the articles. At this stage, the choice of further articles used in the master thesis was made. This step is the most voluminous and complex, as the amount of information grows exponentially every year [4]. The results of the literature review are presented below.

Data is a vital driver of the digital economy [5]. Before going directly into the master data management systems, it is worth defining the data in general.
The five key types of data are presented below:

1. Metadata.
2. Reference data.
3. Master data.
4. Transactional data.
5. Historical data [6].

Metadata is data about data. It is needed to understand and determine what data the company operates. Metadata defines structures, data types, accesses. There are various schemes for describing metadata. For example, an eXtensible Markup Language (XML) Schema Definition (XSD) can describe the structure of an XML document.

Reference data is relatively infrequent data that defines the values of specific entities used to perform operations across the enterprise. Such entities most often include currencies, countries, units of measurement, types of contracts, and accounts.

Master data is the underlying data that defines the business entities that the enterprise deals with. Such business entities usually include (depending on the subject industry orientation of the enterprise) customers, suppliers, products, services, contracts, invoices, patients, citizens. In addition to information directly about a particular master entity, master data includes relationships between these entities and hierarchies. For example, it can be essential to identify explicit and implicit relationships between individuals to find additional sales opportunities. Master data is distributed throughout the enterprise and is involved in all business processes. Usually, master data is perceived as a key intangible asset of an

enterprise since its quality and completeness determine the effectiveness of its work. In Russia, the term “normative reference information” is often used instead of “master data.”

Normative reference information is a regular part of business information, knowledge about the objects and subjects of business entities included in the circle of interests.

Transactional data is data that is formed as a result of an enterprise performing any business transactions. For example, for a commercial enterprise: sales of products and services, purchases, receipts/debits of funds, landings into the warehouse. Usually, such data is based on the Enterprise Resource Management system (ERP) or other industrial systems. Naturally, transactional systems make extensive use of master data when executing transactions.

Historical data is data that includes historical transactional and master data. Such data is often accumulated in Open Data Source (ODS) and Data Warehouse (DWH) systems and is used to solve various analytical problems and support management decision-making [6].

2.1 Master Data Management as a discipline

Master data contains vital information about a business, including customers, products, employees, technologies, and materials [7]. Master data is specific in that it is relatively rarely changed and is not transactional. In some cases, master data supports transactional processes and operations, but it is used for analytical activities and reporting to a greater extent. We can say that master data defines the enterprises themselves. It represents all the components of the business, and it is a complete representation of the enterprise.

The concept of Data Governance (DG) is a description of methods for solving high-level strategic tasks of a business for working with data [8].

Data governance includes:

1. Methods for evaluating data as an asset.
2. Ways to organize data management processes.
3. Ways to create data management policies and regulations.

4. Roles of specialists in the data management process.
5. Other recommendations for the exercise of leadership and control over information assets [9].

DG class solutions are a set of tools aimed at creating a unified view of business information assets and ensuring high data quality throughout their entire lifecycle. It is essential to distinguish between Data Governance (DG) and Master Data Management (MDM). MDM class solutions perform an executive function. They allow organizing the practical implementation of data management: quality assurance, standardization, validation [10]. DG class solutions perform a supervisory function. They allow to create requirements for data management and monitor their implementation: the formation of business terms, linking business concepts with the technical implementation of the IT landscape, defining requirements for conducting MDM [11].

The governance and management of core data are often implemented at the business development stage when data problems cause significant damage to the business in monetary, reputational, or other terms. Moreover, the transition to data management is designed to solve one or several business needs and tasks at once. An example of such a task is compliance with regulatory requirements in highly regulated industries, such as the financial and banking sector [12].

Depending on the business environment, the following data management implementation scenarios are possible:

1. Without an MDM system. This scenario creates requirements for data management, including developing business terms; analyze the quality of data in different systems; regulates the work with data; unify reporting.
2. With an MDM system. This scenario expands data source management capabilities, allows the application of data management requirements directly to the MDM system, collects new reports, and so on. Thus, using MDM and DG solutions enables the synergy and mature management of crucial business data [13].

If the business is already using an MDM system, additional integration will be required. If the use of an MDM system is only planned, then the implementation of MDM and DG will require significant labor, and the best choice is to use a comprehensive solution that combines both solutions. The main stages of implementing a DG solution may consist of

analytics, where the strategy, goals, and objectives are defined. A business examination identifies essential details, opportunities, and threats. Next, designing a business solution where the data management structure, principles, and policies are defined, requirements are evaluated. Basic entities, business terms, dictionaries, reports are created during the development of a solution. Finally, suppose implementing the solution, where organizational changes and problematic issues are managed, embeds data management in processes and coordinates data management participants' departments. The implementation methodology will be covered in chapter four.

Next, we will look at two cases where DG and MDM should be applied.

2.2 Case of monitoring the protection of confidential data

The set of data specified in the business rules is defined as confidential information, whereas the data from this set is not confidential. For example, such a set can serve as information about critical customers: personal data, contacts, order history, personal discounts. Sensitive data is stored in a separate database. Various business services send requests to the confidential database to retrieve individual data. For example, the delivery service requests only the name, address, and list of items in the warehouse. Since the business is large and many different systems are used for their intended purpose and implementation, there is a risk of data leakage. Similar situations occur when responses to requests accumulate in an unprotected place in the IT landscape. For example, in the logs of one of the systems. The MDM and DG class solution allows identifying where confidential data sets still occur and, if necessary, tracking the full path of technical data placement: find out the origin of the data and the relationships between other sources and determine the initiator of changes at each stage of the movement. With this, it is possible to eliminate security flaws.

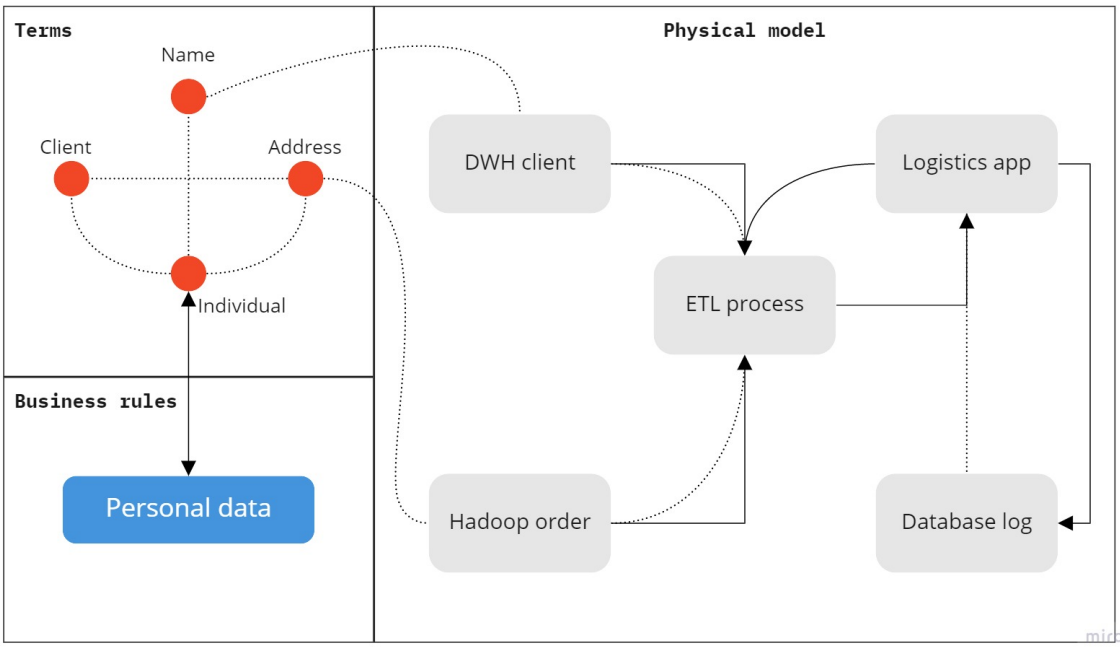

Fig. 1. Identifying unprotected sensitive data.

Figure 1 shows a case for identifying unprotected confidential data. Business rules define that the "individual" object with the attributes Full Name, Address, and Client is confidential data. Different systems request either the address or the full name separately, but in the logistics department, the attributes of the individual are collected again. At the same time, in the database logs, confidential data is unprotected, which is a threat.

2.3 Case of working with reports

Problems with reporting can occur for various reasons. For example, some issues can be:

1. Inconsistency of data within several information systems, different divisions of the business.
2. Changed requirements of regulators.
3. Business acquisition and related differences in understanding of business terms and critical indicators.
4. Problems of aggregation of different units of measurement.
5. There is a need for unified and reliable data sources and a shared understanding of all business terms to create new reports [14].

MDM class solutions combine all existing data models into one single model, based on which all conflicts are closed in units of measurement, business terms. When conditions or data requirements change, adjustments are made to the unified data model, which automatically moves down the data structure. MDM class solutions cover three main scenarios for working with reporting. The first one is to adjust existing reports. A single model can be quickly changed to meet new requirements, unify data, check data in different systems, and search for errors. The second is creating recent reports. MDM solves the problem of finding relevant data and linking the data to the core business concepts. New reports are collected from existing data. For recent reports, it can be used ready-made data sets described in business terms or business rules. The last one is translating new reporting requirements and monitoring the actual execution of reports in each specific information system. Conditions are sent to business units in the form of business terms. Departments generate local reports, which can be monitored for implementation.

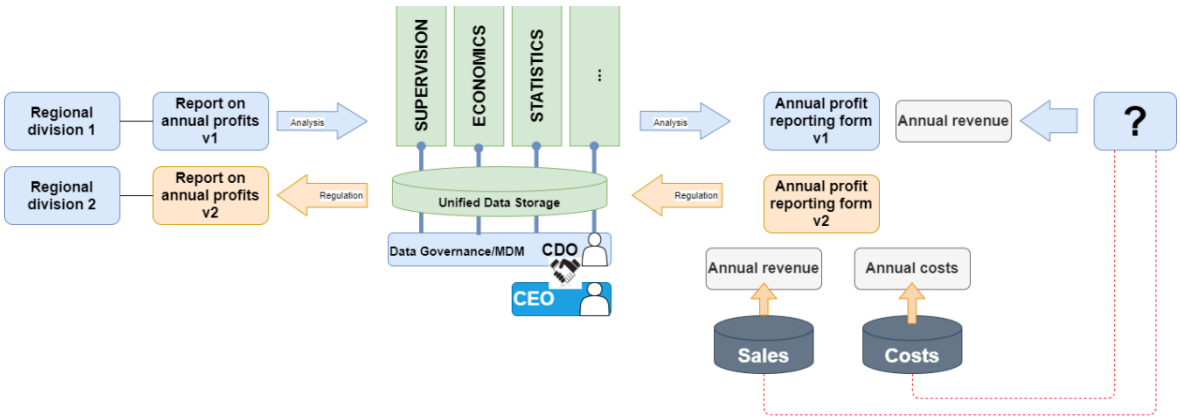

Fig. 2. Regulation of annual profit reporting.

Figure 2 shows the case of regulating the reporting of annual profits. From regional division 1 comes the annual profit report, which, based on the analysis results and comparison with established business criteria, contains data only on annual revenue. The Regional Division 2 report contains both annual revenue and cost data. Since the two reports differ in the basic understanding of the business term “Annual Profit Report,” it is necessary to regulate this concept, which will entail adjusting one of the reports.

2.4 Master Data Management Systems

Before moving on to the master data management system, it should be defined what master data management in general. Master Data Management (MDM) is a discipline that works with master data to create a "golden record," that is, a holistic and comprehensive view of the master entity and relationships, a master data reference that the entire enterprise uses, and sometimes between enterprises to simplify the exchange of information. Specialized MDM systems automate all aspects of this process and are the "authoritative" source of enterprise-wide master data. Often, MDM systems also manage reference data.

The methods of using MDM determine what the MDM system will be used for in the enterprise or who will be the consumer of the master data.
The main methods of use are three:

1. Analytical.
2. Operational.
3. Collaborative [15].

The analytical usage method supports business processes and applications that use master data to analyze business performance, provide the necessary reports, and perform analytical functions. It is often done through the interaction of MDM with BI tools and products. Usually, an analytical MDM system works with data only in read mode. It does not change the data in the source systems but is engaged in cleaning and enriching them.

The operational use method allows the collection, modification, and use of master data during business transactions and maintains the semantic consistency of master data within all operational applications. In fact, in this case, MDM functions as an Online Transaction Processing (OLTP) system that processes requests from other operating applications or users. This mode often requires building a single integration landscape using Service-Oriented Architecture (SOA) principles and using the Enterprise Service Bus (ESB) tools. It is ideal if such tools are either directly part of the MDM system or its continuation. There are vendors with MDM and ESB solutions in their line that are deeply integrated [15].

The collective use method allows the creation of master entities in cases where collaborative interaction between different user groups is required during this creation. Such coordination usually has complex "branching" business processes consisting of various automatic and manual tasks. Various data specialists perform manual tasks in the order defined by the

business process. Most often, the collective use method is used in the product domain. For example, several people are responsible for entering different data, manual work, and final approval when creating a new product. The MDM system must allow configuring custom business processes to support a particular enterprise's business processes quickly.

Modern MDMs, in addition to the master data storage service, the only source of truth, usually include a whole set of services: Extract, Transform, and Load (ETL) services, data quality management services (profiling, standardization, enrichment, deduplication), metadata management services, access control services, services for the work of data experts, hierarchy management services, search services, and many others [16].

The purpose of MDM is to provide a holistic view of the main components of the business [17]. Any company, as a single entity, needs to be informed about itself. IT professionals do not have to deal with information but with data that represents or contains information. Almost universally, two terms, "information" and "data", are identified, which is incorrect. In real life, what is required is not general reasoning about information but solutions based on intuitive principles, and MDM is one of them. The concept of MDM is uncomplicated. MDM is operating with data that is distributed between different subsystems that are available to other users. This data can be merged into one so-called "reference" or "master file" in the simplest case. Such a file can be, for example, a customer file that is created and used by different departments. Sometimes an alternative name is used for MDM – Reference Data Management (RDM). Thus, the main goal of master data management is to ensure that there are no missing, duplicate, incomplete, or inconsistent records about business domain objects in all corporate information systems [18]. In the next section, data quality practices in the MDM systems will be covered.

2.5 Data quality

Data quality is a vast and increasingly relevant topic in today's world. Different authors define the term data quality differently. Some claim that this is the degree of suitability of the data for a specific use [19], [20]. Others emphasize that this concept is multidimensional and consists of accuracy, completeness, and other criteria [21]–[23], [24], [25]. Data quality assessment is the first and significant step in a time-consuming process called Data Quality Improvement.

Over the past few decades, various methods for assessing data quality have been developed [26] [27]. Most of them are related to the relational data model and are based on analysing individual values without using other tables. The exception is the method of cross-domain analysis, which allows handling redundancy and inconsistency of data in several tables [28]. In this section, a data quality assessment method is introduced based on the comparison of several sources. This approach allows determining the quality of a data instance in the context of various criteria and with the use of multiple metrics for evaluation.

Data is the fuel for artificial intelligence systems, the raw material for analytical algorithms, and the basis for business process automation systems [29]. If decision-makers do not have timely, relevant, and reliable information, they have no choice but to rely on their intuition. Data quality becomes a crucial aspect [30]. This section aims to describe what requirements and indicators are applied to data and help define the data's trustworthiness. Then it is explained why data quality matter and is essential in business and the digital economy.

The overabundance of various data and the abundance of multiple tools for working with them can be misleading: it may seem that to monetize data and increase employee productivity, it is only enough to invest in advanced tools, machine learning, business intelligence tools, which, for example, allow to develop individual attractive offers through a deep understanding of the market and consumers. Nevertheless, Big Data (3V: Variety, Velocity, Volume) is worthless without Veracity. The quantity, speed of collection, and variety alone do not guarantee an array of high-quality, workable data. Moreover, as numerous surveys show, the excess of data causes stress for employees, and the diversity of information, the disparity of its sources, and the lack of standardization are vital factors that prevent companies from gaining new knowledge from data.

The data is worthwhile only when the business can extract valuable business information from them [31]. Data quality is a characteristic of digital data sets that shows their suitability for processing and analysis and compliance with the mandatory and special requirements imposed on them in this regard [32].

The data quality is applied to the following objects:

1. Attribute values. The data contained in the attribute of a particular look-up object. For example, the attribute is Country name, attribute value – Russia.
2. Data blocks. The values of an attribute or group of attributes that describe a single entity included in the description of the look-up object. The peculiarity of the attributes that are included in the same tuple is that they change together. For example, the passport attributes change together when the document is changed.
3. Object record. A set of attribute values and data blocks is united by an ordinary description object associated with it — for example, data about an individual, including its details and documents.

Harmonization and validation of the data should be used to ensure data quality for later uploading and working [33]. Moreover, the rules should be described to find duplicate records, create a "gold" record, define the rules for combining duplicate data, and describe the system settings that need to be implemented within these requirements [34]. Thus, data quality is often described as a concept with multiple dimensions [35]. Over the years, a wide variety of dimensions and dimension classifications have been proposed [36], [37] [38], [39]. In particular, the ISO 9000:2015 standard defines the quality of data by the degree to which it meets the requirements: needs or expectations, such as Completeness, Conformity, Accuracy, Consistency, Integrity, and Timeliness [40].

Completeness is defined as expected comprehensiveness: the property of information that exhaustively characterizes the displayed object or process. As long as the data meets the expectations, then the data is considered complete. Conformity refers to information matching an internal or external standard. The accuracy of information is determined by the degree of proximity to the actual state of the object, process, phenomenon. The value of information depends on how important it is for solving the problem and how much it will be used in any future activities. Consistency is about compliance with established (reference) data. It refers to whether the data match information from other sources. Consistency determines its reliability. Integrity refers to the completeness of the data reflection of the natural state of the target object, which shows how complete, error-free, and consistent the data is in terms of meaning and structure while maintaining their correct identification and mutual connectivity. Timeliness is the ability of information to meet the consumer's needs at the right time and receive data at a reasonable time [41],[9], [42].

The most important indicator of data quality is its integrity. It has a substantial impact on data compatibility and manageability. Furthermore, repeated publication of data with a violation of integrity will necessarily affect the trust in their provider. Data integrity is not something separate from meaning, structure, or format and must be respected at all levels of digital information [43].

Data integrity violation is possible at different levels:
Semantic level – when collecting, an error was made in the completeness or recording of the data so that the meaning becomes incomprehensible that such data describes [44].
Structural level – when ordering data elements or processing data, an error is made in the completeness, recording data so that a part or an entire structure becomes "incomprehensible" [45].
Notation level – when writing, storing, or reading data, an error is made to convert individual digital data elements or write them together so that it is impossible to correctly establish separate individual units and relationships between them in the data [45].
Schema level – when writing, storing, or reading data, an error is made at the logic or format of individual digital data elements or their relationship, so it is impossible to extract meaningful information about the subject area from the data [46].

To keep up with the development of digital trends and get value from data, data needs to be managed, checked for their behavior in new conditions and systems, and monitored for relevance, sufficiency, and relevance.

Most often, we talk about the data quality of attribute values. However, the main task of DQ and MDM is to ensure the integrity of object data and its comparability between multiple systems.

The data quality of an object consists of the data quality of individual data blocks and their attribute values. In general, based on the tasks of MDM projects and their implementation styles, the role of DQ varies greatly.
The four existing styles of MDM implementation are presented below [47].

The general catalogue is working with the reference information. It is only needed to control the quality of data entered by operators and if there is a primary input or loading of data, then comb this data automatically. This implementation style is often used at the initial stage of implementing MDM solutions and reference information methodology. Applicable only for the reference information with a low degree of variability and a low rate of change. It allows concentrating all data changes in one place, thereby avoiding errors in entering particularly critical data.

Data quality modules are not used when using this style since the only reference external classifiers are used. Classifiers are published either without changes or based on the established conversion rules.

The analytical implementation style already requires a full range of DQ work, including searching for and combining duplicates, but the data quality level requirement is relatively low since 80% of the quality level is sufficient for building analytical reports and models. This style is often used when using the " the reference information + master data" technique and allows collecting a standard set of data for internal reference information and master data. Then, a partial update of the condition data is applied. In most cases, the "replicate when modified or added" condition is used.

Using this implementation style, data quality modules are most in demand at the consolidation stage to create a common database and use a decentralized directory management scheme with one or more data source systems. Thus, data quality requirements are high at the consolidation stage and medium at the real-time data processing stage. At this stage, the front-end systems strengthen the control of input data and implement a service model of control procedures.

For the harmonizing style of MDM implementation, the same set of DQ is already required as in the analytical one, but with a quality level of 95%. In addition, the style implies the "alignment" of the reference information and master data in the recipient systems at both the reference information and master data levels. It means rechecking all data, including transactional data from previous periods. The implementation style and alignment procedures require a high level of project maturity and tested data quality technologies.

Special conditions are also imposed on the MDM system to store the history of changes and restore the values of the reference information and master data at any time.

Furthermore, the transactional style of implementation does not tolerate errors at all since automated business processes are imposed on it. The style implies the use of standard identifiers for the reference information and master data. Cross-system exchange is performed using the same identifiers in transactions without transmitting significant reference information and master data. It allows to organize guaranteed data exchange between systems and avoid intersystem failures at the data integration level. It also solves the problem of personal data transfer. Since the use of the merge, the transfer of personal data does not occur, but only system-wide identifiers for personal data are transmitted.

An intermediate version can be used with MDM system cross-link tables, and intra-system identifiers carry out the exchange between the systems. The method is used as an intermediate option for systems that do not have the possibility of improvement in terms of working with a single identifier. The implementation style requires exceptionally high-quality data and mechanisms that exclude the possibility of error both at the level of automatic decision-making and the human factor.

The above implies the need for continuous data quality control on both the recipient and the provider sides. It, in turn, forces the development and use of unique control and measurement tools.

2.6 Data quality practices

This section aims to describe the general functional requirements for the quality of look-up entities' data, what can be done to fix the errors and defects in the data, and how to manage data quality. Technical error is an error (typo, grammatical or arithmetic error, or similar error) made by the user in the implementation of data and led to a discrepancy between the information contained in the system and the information contained in the documents based on which the data was entered in the system.

While a tremendous amount of research is devoted to schema transformation and schema integration, data cleanup has received only a tiny amount of attention in the research

community. Several authors have focused on the problem of identifying and eliminating duplicates, for example, [48], [49], [50], [51], [52]. In addition, some research groups focus on general problems not limited to but related to data cleanings, such as unique approaches to data mining [53], [54] and data transformations based on schema matching [55]. More recently, several studies have proposed and explored a more complete and uniform approach to data cleanup, covering multiple transformation steps, specific operators, and their implementation [56], [57], [58].

If we consider operations with data quality, then the first thing is always data profiling. This is primarily an assessment of the data structure of the sources, the possibility of mapping them to the target model. Profiling includes evaluating the essential quality criteria, fullness, length, uniqueness, pointers to reference directories, then their composition and inconsistency, the ability to combine them with the target reference directories. The general purpose of profiling is to understand what the data is, how complete it is, how it can be transferred to the target model, and how it will need to be transformed. [59]

Metadata is primarily the structure of the source data. Therefore, the closer we are to the data, the better. Likewise, the fewer transformations the client does with the data, the better because any data transformation without DQ is always a loss of part of the data.

Then, when the data is already loaded, the overall quality of the loaded source data should be evaluated.

Validity is an indicator of the quality of an attribute value, tuple, or record. Validation is performed twice in the data lifecycle, before the cleaning procedure: standardization, transformation, harmonization, restoration and after these procedures [60]. The first validation result depends on what techniques will be applied to get the best data as quickly and cost-effectively as possible. After cleaning, the quality of the final data should be reassessed. The results of this evaluation are already used in object matching and merging operations.

The validity of the attribute value is set to one of the values:

1. Critical.

2. Risky.
3. Reliable.
4. Guaranteed.

Critical validity indicates that the content of the attribute value contains an error that is incompatible with the business use of the data. Business use includes matching values for subsequent deduplication. Risky validity indicates the presence of errors that do not affect business use. Reliable validity indicates that the controls check no errors. Guaranteed validity indicates the business sense content in the attribute value.

Each attribute has two validity characteristics. The following types of validity indicators are set for the attribute value by length and content. If several controls use a single indicator, the worst value is selected, except in guaranteed validity. If the control guarantees validity, then the other controls lose their values.

Each tuple has one validity metric. It is calculated based on the values of the validity indicators of the attributes included in the tuple. The calculation involves indicators of the validity and significance of the attribute.

For example, in the Tuple Document of an individual, there are attributes:

1. *Document status.*
2. *Document type.*
3. Last name.
4. Name.
5. Gender.
6. Date of birth.
7. *Document series.*
8. *Document number.*
9. Date of issue of the document.
10. The issuer of the document.
11. Code of the department that issued the document.

Attributes highlighted in italics are used in the company's business processes and are analyzed in automatic processing, while other attributes are used only in printed forms, but they are not processed automatically.

In this way, attributes can be separated by importance. Three states can determine the character of the importance of an attribute. The key attribute is an attribute that carries the primary business meaning of the tuple. When this attribute is changed, the tuple is recognized as unique. It is used in the company's business processes, processed automatically, and determines the meaning of the entire tuple. A significant attribute is an attribute that carries a vital business meaning but not a key one. The loss of the attribute significantly affects the information content of the tuple. Used in the company's business processes, processed automatically, does NOT determine the meaning of the entire tuple.

The additional attribute does not carry a significant business meaning in the tuple. The loss of the attribute is not critical for the business meaning of the tuple. It is used in the company's business processes, is NOT processed automatically, and does NOT determine the meaning of the entire tuple.

Each tuple has its formula for determining the validity of the tuple. Nevertheless, the general idea is that the validity of the tuple is determined primarily by the validity of the key attributes. In the future, the validity of the tuple is used in the merge and conditional publication operations. The choice of the best possible option is based on the quality of the data and its relevance. However, individual attribute metrics do not allow the selection of the entire tuple.

Table 1 shows examples of validation on different types of data.

Table 1. Validation examples.

Name	Description
Presence of double hyphens	The attribute value must not contain two or more consecutive hyphens. If the requirements are not met, the attribute value is set to the critical content validity. If they are met, reliable content validity is set.
Validation of numeric attribute values	Validation of numeric attribute values by format control is to check whether the value is a number.
Validation of uniqueness	Validation of the uniqueness of an attribute value consists of finding an equal attribute value among other objects. Implementing uniqueness cannot be implemented through a database function since it is always possible to get an exception.
Cross-validation	In addition to validating a single attribute value against a list of valid values, dependent checks can be performed on the values of several attributes of the same tuple or different contours. Similarly, the comparison can be made both for open lists and for closed ones. For instance, they are determining the correctness of the Gender and Surname.

Standardization removes part of the characters from the attribute values to increase the information content. For example, it can be rough cleaning, such as removing extra spaces, double hyphens, and dashes, removing invalid characters. In general, standardization refers to operations with symbols of the attribute value without considering the symbol's meaning [61].

In most cases, standardization repeats the validation conditions and removes characters that do not match the condition.

Table 2 shows examples of standardization of different types of data.

Table 2. Standardization examples.

Name	Description
Multiple spaces	The attribute value must not contain two or more consecutive spaces. All spaces that are more than one in a row should be deleted.
Format standardization of numeric attribute values	Standardization of numeric attributes by format attribute consists of converting the attribute value to the accepted number format. The bit depth, precision, and other attributes of the number are set. If bit separators, such as dots, spaces, and others, are used, they are removed, except for the decimal part separating them.
Standardization of date type attributes	Standardizing the values of the date type attributes by format is to reduce the characters of the text string to a single alphabet and write (the dominant alphabet, all uppercase) if the input of the function receives the attribute values as text and not as a date or date-time. Conversion from text to date or date-time format is described in the section harmonization.

The transformation works with part of the attribute value. Transformation is a procedure for changing the value of an attribute based on the specified change rules to increase the information content. [62] For example, transformation brings it to a single alphabet, English or Russian, replacing non-standard abbreviations with standard or full ones (kg on kilograms or cm3 cubic centimeters). In general, the transformation includes operations with a part of the attribute value and changing it to another one. An example of transformation is replacing characters in the text and numeric attributes. All characters that do not match the valid ones are converted through the substitution tables. The replacement is made in the dominant alphabet of the attribute value unless otherwise specified in the settings. If there is no match in the substitution table, the symbol is deleted in the standardization procedure performed after the transformation [60].

Harmonization operates on the entire attribute value and not part of it. Harmonization is a procedure for bringing attribute values to a specified storage format. For example, it can be

converting dates to a single format (Date), replacing attribute values through transcoding tables (converting them to reference look-up entities), converting numbers from strings to a numeric format, and converting them to fundamental units of measurement. In general terms, everything that provides the ability to store the value in the desired MDM format is referred to as harmonization [63]. Text attribute values can be harmonized across open and closed transcoding tables. Transcoding tables are most often used when the MDM model has enumerations, dependent look-up entities, and links to other entities. The difference between harmonization for open transcoding tables and harmonization for closed transcoding tables is that if there is no value in the open transcoding table, the total value does not change since storage allows other values, and when the transcoding table is closed, the value "Undefined" is assigned. A link is made with the enumeration value "Undefined," the reference value "Undefined" is not connected with the dependent entity.

Data recovery is the procedure for mapping an attribute value or group of attributes to an etalon. The recovery result should be a pointer to which the record in the etalon is the input set. For example, a typical representative of recovery is parsing a mail address. Recovery has an error rate: False Acceptance Rate (FAR) and False Rejection Rate (FRR) [64].
The first kind (false positives) error is when there is no identification of the record. Analog from FRR biometrics is the probability that a person may not be recognized by the system (false access denial rate). The second kind of (false negatives) error is when there is the false identification of the record. Biometrics analog FAR (False Acceptance Rate) is a percentage threshold determining the probability that one person can be mistaken for another (false access coefficient). It is also referred to as the error of the second kind. Each recovery procedure is designed for an object or attribute. The general rules describe only the methods that can be applied during development. The use or non-use of standardization, transformation, and harmonization before the restoration procedure is determined for each attribute individually. Data can be restored data based on non-obvious data present in the record. For example, the gender can be restored by full name in most cases. Recovery by external etalons can be carried out using various algorithms that identify the attribute value with one of the values of the external etalons. All of them are aimed at determining the object in the reference as accurately as possible.

The main idea of recovery is to identify a specific record in the reference databases corresponding to the original record from the data row. The most challenging part of recovery is deciding which of the appropriate records is the final one in the recovery process. Therefore, it is vital to have as complete a knowledge base as possible. If there is no record in the knowledge base, then deciding on an undiscovered record in the presence of very similar ones is a separate difficult task.

In an ideal world, recovery mechanisms can determine a record from several knowledge bases and decide which of several records from several knowledge bases to accept as the final one or not to accept more than one and return not found.

Finally, the last data quality rule is matching or search for duplicates. The purpose of the matching procedure is to determine the duplicates in the selection. The result of the matching procedure is the pairs of duplicates found. Each mapping must have one of three results – "Two objects are duplicates," "Two objects are possible duplicates," "Two objects are not duplicates." It is important to note that objects with a common duplicate are duplicates, no matter how similar. The same tools can be used for mapping as for recovery procedures. The FAR/FRR characteristics also apply to the matching procedure.[48], [49], [62]

When matching, a complete search of the "each with each" selection and matching only the most suitable objects for comparison can be used. The selection of the most suitable objects for comparison is carried out through the clustering procedure. Its own rules determine the allocation of a group of objects (clustering) for entering an object into the cluster.
The comparison can take place according to several rules and by different attributes.

The primary use case compares objects of the same entities, look-up entities, but there are cases of matching objects by classifiers with intersecting attributes on branches. A frequent case during the initial download is comparing different entities, look-up entities, and classifiers.

In most cases, the values of an attribute with critical validity are not allowed to be compared. For attribute values with validity lower than "Guaranteed" or "Reliable" in most cases, the

result of matching should not be set - “Two objects are duplicates,” maximum “Two objects are possible duplicates” since matching on erroneous data often leads to erroneous decisions. Since the number of comparisons of two objects according to several rules can exceed two dozen, then on large amounts of data, objects that have not changed since the last comparison can avoid re-matching by having a separately stored attribute of the pair – “Are not duplicates.” In this way, re-mappings will only occur on those objects that have been changed since the last mapping. For example, object mapping compares sets of attribute values, complex attributes, and relationships between two objects. Simple object matching is done by comparing the attribute values specified in the rules. Matches are made only in pairs. Multiple objects are also compared in pairs. In situations where object A is a duplicate of object B, object B is a duplicate of object C; it is fair to say object A is a duplicate of object C [16].

The overall process is presented in Figure 3.

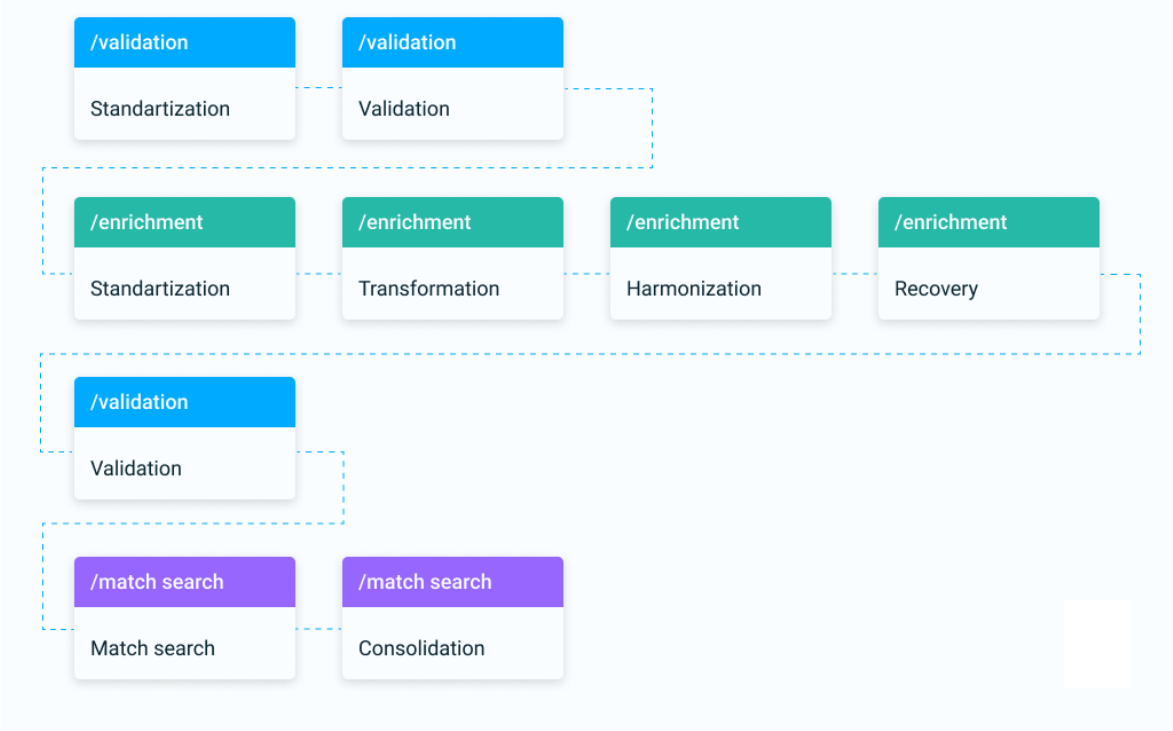

Fig. 3. The sequence of the DQ operations.

This section aims to study the current trends in the management of master data and how the highest data validity can be achieved. Two research questions were answered through the whole section:

RQ1. What are the existing solutions of master data management systems?

RQ2. What are the existing practices of data quality?

3 THE APPROACH TAKEN TO ANSWER THE RESEARCH QUESTIONS

The choice of the qualitative approach in current research falls based on several reasons.

Firstly, the qualitative approach is used when there are human interaction and behavior, and it aims in-depth of its understanding. For instance, in software engineering, it is not easy to research without considering social interaction. For example, we are interested in factors of how the project management methodology influences the development team's work.

Secondly, a qualitative approach is used when it is necessary to answer why and how questions – in addition to what, where, when, and how many/how often.

Finally, the qualitative approach is used when building the theory since it is based on induction reasoning. For example, in the article by Kathleen Eisenhardt, she recommends building the theory using a case study method. She mentions that the case study is particularly relevant in new topic areas [65].

Overall, the qualitative approach attempts to interpret words, perceptions, feelings rather than analyze numbers in contrast with the quantitative approach. This research aims to study how the development of the data quality subsystem influences the work of the companies, what are the benefits of using data quality management, and why data quality is essential in the industry?

It is generally accepted to use qualitative research to study the features of human interaction and consider social issues. As any software development includes human interaction, it is reasonable to use a qualitative approach.

There are many approaches to conducting qualitative research.

The research method is a single-case study and an empirical inquiry. It has an induction nature. As with any research in the case study, the researcher first identifies the problem and forms the research questions. Drawing up research questions allows not only to determine

the direction of research, the necessary resources, and the limitations [66]. The primary purpose of the research questions is to build a theory. In this regard, questions should provide an opportunity to study the phenomenon in depth [67]. A case study studies contemporary phenomena in their natural context, including people and their interaction with technology. As part of the research, it is also planned to study outcomes from developing the systems and practices specific to the company [68].

In this study, an observation and document analysis will be used as a data collection tool. Observation is a method of collecting primary empirical information about the object under study utilizing systematic and direct visual and auditory perception. Significant social phenomena, processes, and situations, that are subject to control and verification of the study, are recorded to generate outcomes. An observation is s first-degree technique, as the researcher is in direct contact with the subjects and collects data in real-time [69]. Document analysis is a systematic procedure for reviewing or evaluating documents. Document analysis uses information recorded in a handwritten or printed text, computer, and other information media. The advantage of the document analysis method is that it opens up vast opportunities for understanding the natural phenomena reflected in the documentary sources about the activities of the company and the project team. It is a third-degree technique as the researcher conducts an independent analysis of work artifacts already available, and sometimes compiled data is used [69]. A severe problem with document analysis is the lack of confidence in the document's reliability and content.
Table 3 shows the research questions and the research method used to answer the question. With the question, the sections that relate to it are also given.

Table 3. Research question and research methods.

Research question	Method and sections
How to develop and implement a data quality assurance subsystem?	Case study. Sections 5 and 6
RQ1. What are the existing solutions of master data management systems?	Literature review. Section 2
RQ2. What are the existing practices of data quality?	Literature review. Section 2
RQ3. How should the data quality assurance subsystem be developed and implemented?	Case study. Sections 5 and 6
RQ4. How effective is the development and implementation of a data quality assurance subsystem?	Case study. Section 7

These instruments allow broad information about the topic under study and get a deeper understanding of the issue [70],[71].

4 PLATFORM DESCRIPTION AND JUSTIFICATION OF THE NEED TO CREATE THE SUBSYSTEM

The chosen company for the case study specializes in developing and implementing enterprise master data storage and management systems. The group includes several companies, each of which is engaged in a specific area, but all of them, in one way or another, are directly related to high-load systems in the field of data management.

The purpose of the chapter is to study the core of the company's regulatory reference information management platform and introduce the basic features of the existing platform.

4.1 Purpose and capabilities of the platform

The data management platform is based on the state-of-the-art free software technology stack. It has received many positive reviews from well-known analytical agencies such as Gartner and Forrester. Their users include primary transport, energy, telecommunications companies, educational institutions, industrial enterprises, and various public administration institutions.

The company's methodology for implementing the platform is based on the international DMBOK standards, fully adapted to the realities of the Russian market, and equipped with a robust set of industry modules. The unified product development roadmap is based on current industry trends in Data Management & Governance, Data Quality Assurance (DQ), regulatory reference information, and new technologies and methods for processing large amounts of data [64].

The platform is designed to build centralized data management systems, including critical business data and regulatory and reference information.

The main functions of the MDM platform include:

1. Data processing. Support for creating, searching, viewing, editing, and deleting records.
2. Administration. The platform's toolkit for managing users and roles.
3. Data management.

4. Integration. A set of different APIs for integration with external systems.

Often, master data management is implemented at the stage of business development when data problems cause significant damage to the business in monetary, reputational, or other terms. Moreover, the transition to data management is designed to solve one or several businesses' needs and tasks at once. Data management in the platform includes: create, view, and edit a data model, setting up data quality rules, duplicate search rules and consolidation rules, manage data sources, and view a library of data cleaning and enrichment functions.

All the functions of the platform are divided into two main user groups. Each group performs its tasks.

Data operator. The main task: processing data that represents individual records of entities or look-up entities. For example, the platform can create an entity of "Developers," where each developer's information is recorded. The record contains several attributes. It can be the organizational and legal form, name; phone number; legal address.

Simple processing tasks of data operator are:

1. Enter attribute values for the current moment and a specific state of the record in the past or future (for a different period of relevance).
2. Edit existing records.
3. Delete records.

The platform allows the creation and manages business processes, organized as an algorithm for approving any change in records. In this case, tasks are created for approving changes.

There are three reasons for changing records:

1. Processing records that are partially or entirely duplicated. The search rules for potential duplicates are configured separately. As a result of the rules, records that have potential duplicates are marked specially. The operator's task is to compare these records manually and, if the two records are duplicates, combine them into one.
2. Processing records where quality errors were found. Data quality rules are configured separately. As a result of the quality rules, records that have errors are marked specially. The operator's task is to open a record with errors, view the attributes that have an error, and correct them. For example, a phone number may have an incorrect format, or the last name may contain numbers.

3. Creating, editing, or deleting records may be related to internal enterprise tasks, such as database expansion.

Data administrator. The main task is to create and configure an information data model and rules for detecting incorrect data. A data model is created to describe the required subject area that contains entities or look-up entities. For each entity and look-up entity, it is needed to create attributes, relations.

The logical structure of the platform is shown in the figure below in Figure 3.

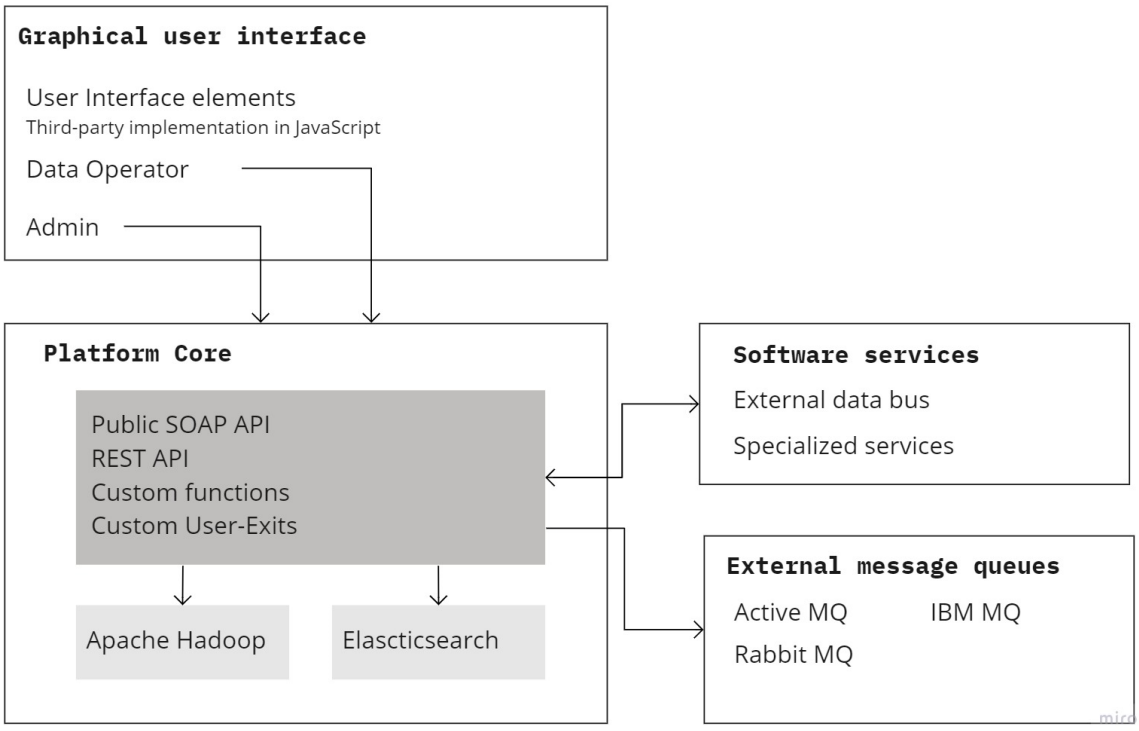

Fig. 4. The logical structure of the platform.

Platform frontend is divided into data operator interface and admin interface. The elements provide access to the relevant functions of the platform for different categories of users. The data operator and administrator interfaces are available at different addresses and have a different set of functions. The platform frontend interacts with the platform backend via a private REST API and runs under the control of a free, open-source Tomcat servlet container.

The order was received for the implementation of the system from a large state customer. The customer is a federal body that performs the functions of organizing a unified state

cadastral registration system. Due to the complex structure of the company and the forced operation with a large amount of data, the company decided to implement a master data management system. The critical factor in deciding on the choice of our platform was the functionality of the platform and the law on import substitution of foreign software in Russia.

Since the essential data indicator is its quality, it was decided to develop an additional module for data verification. The platform's out-of-the-box functionality was not sufficient for fulfilling the customer’s requirements for data quality.

4.2 Justification of the need to create the module with the quality rules

The primary purpose of developing the DQAS is to conduct checks for the presence of technical errors in information and support decision-making within the framework of legal expertise in the implementation of accounting and registration actions in terms of identification and processing of technical errors, including the processing of information received from external sources.

The main goals of creating the DQAS software are:

1. Reduce the time for conducting legal expertise.
2. Minimize the number of unjustified decisions on suspension or refusal of requests by increasing automation of accounting and registration processes.
3. Format a complete and reliable database by improving interdepartmental information interaction with state authorities and local self-government agencies.
4. Improve the quality of system data.

The achievement of these goals is ensured by solving the following tasks:

1. Automation of the process of decision support by Registrars in the implementation of accounting and registration actions.
2. Check the information submitted to the state cadastral registration and (or) state registration of rights to identify and eliminate errors.
3. Check the information for technical errors.
4. Process, check, and enter into the information systems system received from state authorities, local self-government bodies, and other authorized agencies.

These tasks should be solved within the framework of legal expertise, evaluation of external information, and mass verification of the system data for technical errors.

The tasks that are solved during the creation of the DQAS should be considered in the context of the following business processes of the customer:

1. Support of the legal expertise process.
2. Verification of information received within the framework of interdepartmental information interaction.
3. Mass verification of system data for technical errors.

The listed business processes are the object of automation.

4.3 Description of the AS-IS process

The process of verifying information received in the framework of interdepartmental information interaction is described below.

Following the articles of the federal law of the Russian Federation, state authorities and local self-government agencies are required to send documents to the cadastral authorities for entering information into the system, including using the unified system of interdepartmental electronic interaction and regional systems connected to it. The state authority and the local self-government agency are responsible for failure to submit documents. Resolutions establish the list of documents and the composition of the information provided to the cadastral authorities in XML format files.

At the stage of processing information received from state authorities and local self-government bodies, the Registrar must perform the following actions:

1. View the result of the initial document checks for technical errors.
2. View the results of the initial mandatory checks of the source's authority, the format of documents in electronic form, the availability of electronic signatures, and the composition of the sent documents.
3. Determinate the list of real estate objects, the address or description of the location of which has changed due to changes in the passage of borders between the subjects of the Russian Federation.

4. Determinate the borders of municipalities, the borders of a locality, or the list of land plots.
5. Determinate restrictions on the use of which have been established or changed due to establishing or changing the zone's boundaries.
6. Conduct manual checks of the received information and data for inconsistency.

Possible solutions for processing received documents are entering the information and notifying the copyright holder of the changes or completing the application's processing with a refusal if there are appropriate grounds.
In case of a favorable decision based on verifying the received information, the specified information must be entered. In case of an unfavorable decision, the processing of the request is completed, and a notification is generated about the impossibility of entering information.
As seen from the description of the existing process, there is a high probability of human error, missing information, and more resource-intensive document checks due to manual actions.

The object of the case study is the processes planned for automation in the Data Quality Assurance Subsystem (from now on referred to as the DQAS), conducting checks of registry records for technical errors in the information and supporting the decision-making by the Operator in the framework of legal expertise in the implementation of accounting and registration actions.

The purpose of the chapter is to study the processes, form a holistic view of the implementation of the designated processes, identify requirements and proposals for implementing the DQAS to clarify the Requirements Specification for creating the DQAS.
To ensure the assignment of the DQAS following the Requirements Specification, the main features are:

1. Conducting checks for technical errors in the system data.
2. Support for decision-making by the registrar within the framework of legal expertise in the implementation of accounting.
3. Registration actions in terms of identification and processing of technical errors.
4. Processing information from external sources: data model.

5. Data quality criteria and a list of checks carried out within the framework of the development of the DQAS.
6. Steps of accounting and registration actions with the indication of the criteria for their completion.
7. The steps of the legal examination, indicating the criteria for their passage.
8. The required set of data for data quality analysis.
9. The necessary set of information for maintaining the database.
10. Criteria for assessing the quality of the information received through interdepartmental interaction.
11. Criteria for data quality, taking into account further development.

In preparation for the case study, the author collected the regulatory and legal framework, analyzed the sources on the Internet, and formed an examination plan that considers the need to obtain insufficient information on the processes under study.

The examination plan is to interview the customer's representatives of the Departments of Informatization and Development of Electronic Services, maintain the system, analyze the received methodological materials and normative legal acts, analyze documentation, and describe processes.

To ensure the excellent quality of the data, it is necessary to conduct a mass check of the records stored in the system for compliance with the quality rules, such as Format-logical control following the internal rules of data storage of the system, checks for the connectivity of fields, checks for compliance with the requirements of the output formats of external information systems, checks for consistency, the presence of duplicate records.

These checks should be carried out both in a continuous mode, maintaining the quality of the system's data at the appropriate level. According to the specified criteria following the employee's choice is authorized to check the system's information for technical errors.

At the same time, the authorized employee should be able to select a set of records for verification according to specific criteria and choose the rules for conducting checks, the time of the start of the check.

Upon completion of the audit, the user of this process should be allowed to view the audit reports in various sections, using pre-configured reports, or using the report designer to achieve greater transparency of the results of the audits.

The results of the inspections carried out within three days after the discovery should be issued in the form of a protocol of inspections and requests for the correction of technical errors for further referral by the Registrars for correction.
The next chapter will cover the implementation framework.

4.4 Framework choice reasoning

The Subsystem development is carried out using industrial design tools, structural and object-oriented approaches, requirements management tools, configuration management, change management tools, labor intensity assessment and development planning tools, documentation tools, and testing tools. In addition, analysis and modeling tools provide automatic generation of database schemas, prototyping of interface components of the subsystem.
Designing activity processes are implemented using BPM, a system of symbols (notation) for modeling activity processes, and the Bizagi Modeler software tool.

The software development process will be carried out according to the Scrum methodology. For version control of the developed software, the Git system will be used. In addition, to automate the deployment and management of applications in the virtualization environment at the operating system level, Docker software version 1.12 and higher will be used.

Agile is a family of agile approaches based on the values of the Agile manifesto and the 12 principles that underlie it. Since there are many different approaches, we will choose one of them. We will take the most popular project management method for the analysis and consider how the requirement engineering process takes place in it. JetBrains conducted the IT industry research as presented in Figure 4 [72]. More than 40% of developers use the Scrum framework from the agile approach family.

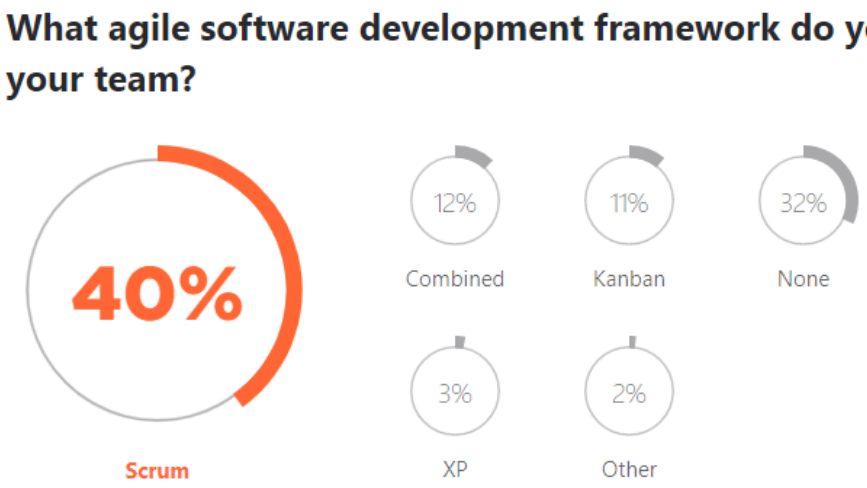

Fig. 5. Result of IT community research. **[72]**

Next, we will look at what this framework is and how requirements management is implemented in it.

Scrum is the method that implements the Agile approach. It is an iterative software development model used to manage the development of complex products. The iteration length is fixed and is two weeks, which allows delivering software regularly.

A key feature of Scrum is regularly reviewing the project content and making changes to the development. It is necessary when developing products that must constantly change under the influence of market requirements.

These iterations are called sprints. At the end of each sprint, team members and supervisors gather to plan the next steps. Scrum defines a set of roles and responsibilities that should not change. Structurally, Scrum provides four operations for each sprint: planning, daily stand-up, demo sprint, and retrospective. During each sprint, the team uses visualizations, such as task boards and charts. It allows the team to track the process in detail and get feedback from the development team. Most Scrum processes are meetings, as this methodology is based on high-quality communication [25].

The process is presented in Figure 5. Product Backlog is a list of functional requirements, ordered by their degree of importance [73]. Requirements can come from different departments: Sales such as requirements of sales, marketing, project Department as

requirements of ongoing projects, Chief technical officer (CTO) as requirements of technological development, support as functionality developed in older releases, or development team as technical and technological debts.

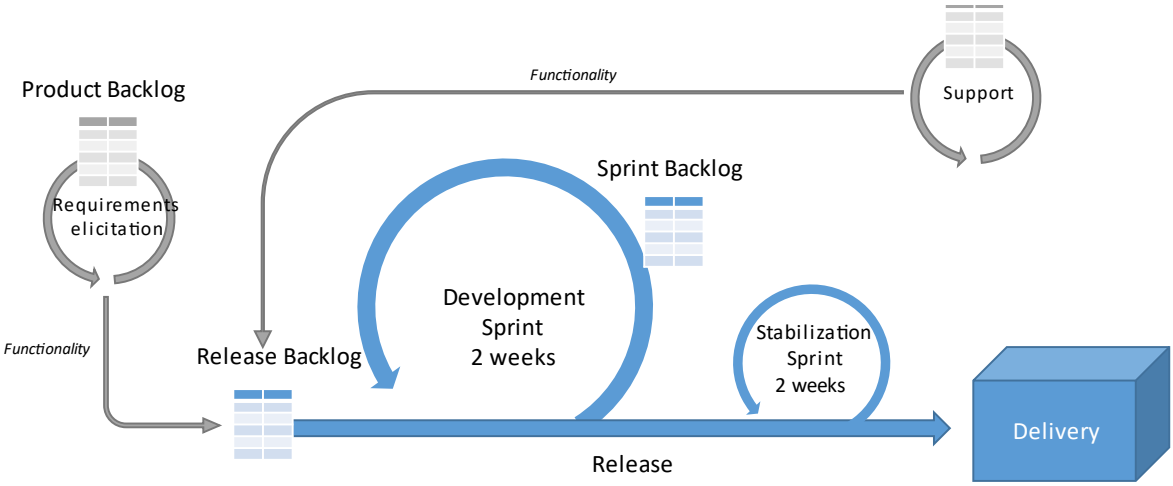

Fig. 6. Release Development: The Overall process.

The release development schedule is shown in Table 3, with participants, duration, and event details.

Table 4. Release Development: Calendar plan.

Stage	**Event**	**Participants**	**Duration**	**Obligatoriness**
Before start	Release planning	Product owner, stakeholders	Two days	Mandatory
	Estimates and release planning	Product owner, team	Three days	Mandatory
Start	Presentation of the actual release volume	Stakeholders, the implementation team		Optional

Table 4 (continuation). Release Development: Calendar plan.

Stage	Event	Participants	Duration	Obligatoriness
Development	Daily stand-ups		15 minutes	Mandatory
	Demo after the sprint	Product owner, team	1 hour	Optional
	Triage rallies (2 hours)	Product owner, team	1 hour	Optional
	Early access builds	Implementation team		Optional
After release	Presentation of the release to implementation team and stakeholders	Product owner, stakeholders, the implementation team	2 hours	
	Retrospective	Team, Product Owner	2 hours	Mandatory

The next chapter presents requirements for the development of DQAS.

5 SYSTEM REQUIREMENTS

The current chapter will be devoted to describing the requirements for developing the data quality assurance subsystem. The main goal of the designed subsystem is to search for data and bulk upload the data needed to check the quality rules.

To maintain the uniformity of the platform's IT landscape, the software of the Subsystem will be implemented in the Java programming language, while PostgreSQL will be used as the DBMS of the Subsystem. The choice of software tools used in the development and organization of the Subsystem is based on the following principles:

1. The software tool must be freely distributed.
2. The program code must be open to allow its modification.
3. The capabilities of the software tool should be well documented.
4. The presence of successful applications of the software tool as complete solutions.
5. Support and use the software solution by many developers, a "platform" for discussion and consultation.
6. Scalability of the software tool.
7. Cross-platform nature of the software tool.
8. The ability of the software tool to work with large amounts of information.

The leading implementation platform of the Subsystem is JavaSE, which is a set of specifications and related documentation for the Java language that describes the architecture of the server platform.

5.1 Requirements for the main functions of the DQAS

The verification module must perform information validation based on the rules configured by the Admin. The verification module should include the following components described below.

A data quality model includes a repository of both existing semantic data checks in the system and newly created checks for entities and information about entities to which the quality rules apply. The data quality model must support the main data types for attributes: string, integer, boolean, date, and graphical data types (any documents, text, and image files

containing applications, scanned documents, tables, graphs, drawings.). In addition, the data quality model must support up to fifty different objects.

The verification should:

1. Provide the ability to search through the components of the Data Quality Model.
2. Allow to import/export a customized Data Quality Model in XML format.
3. Provide access to the Data Quality Model via programming interfaces (APIs) in XML format.

The Data Quality Model Constructor, which is an admin interface for configuring the Data Quality Model, including tools for creating, changing, and loading quality rules used in the Data Quality Model Constructor, the entities of the system, to which the quality rules must be applied, must be defined. This list should be configurable, and changing the list should not require a reboot of the Subsystem.

The Data Quality Model Constructor should determine the order and sequence of applying quality rules, the level of criticality of identified technical errors, and specify and change the description of the error text. The list of processed events (actions with entities of the Federal State Tax Service), upon the occurrence of which the DQAS will launch checks of the objects.

Initially, the list should include a task configured by an Admin or Data Quality Operator. The data quality model should allow the DQAS Admin or Data Quality Operator to create events to trigger checks independently. In addition, the log of the performed check and Federal State Tax Service Reference books should be included.

The logic diagram of the Verification Module is shown below in Figure 8.

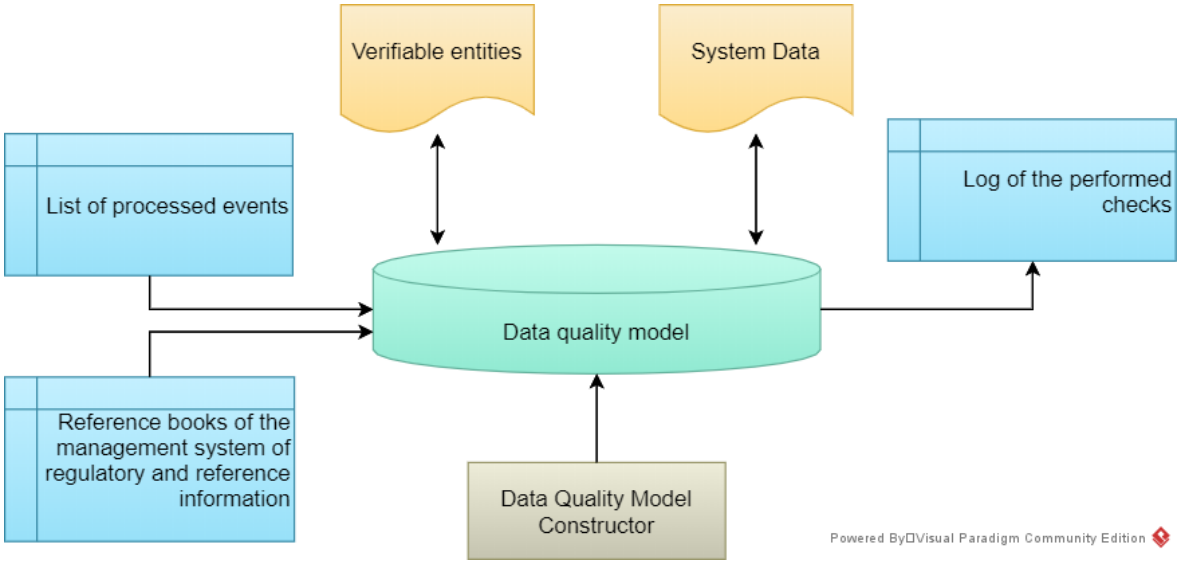

Fig. 7. The logic diagram of the Verification Module.

The verification module is designed to ensure that information checks are performed based on rules that are configured by the admin and will contain the following components:

1. Data Quality Model.
2. Data Quality Model Constructor.
3. List of processed events.
4. Log of the performed checks.
5. Look-up entities.

The DQAS should maintain a history of changes to the Data Quality Model, supporting the versioning of the Data Quality Model in the database, with the ability to compare versions and return to the previous version. Changes made to the Data Quality Model during the configuration process should be applied automatically and should not require a restart of the Data Quality Management System. Changing the quality model should be available without reloading or re-passing the authentication and authorization procedures.

The data quality model should implement the following algorithms for format and logic control for detecting technical errors in the data see Table 4.

Table 5. List of Data Quality Model algorithms for creating quality rules.

Name	Description of the action
Remove Spaces	Removes spaces at the beginning and end of a string
Remove extra spaces	Removes duplicate spaces
Regular Expression	Selects a substring according to the regular expression
Default value	Sets the default value if the input parameter is empty
Format	Formats the string according to the specified pattern
Align to the left	Aligns the row to the left. Adds spaces
Align to the right	Aligns the row to the right. Adds spaces
Parse Date	Converts the string to DATE format
Parse Number	Converts the string to NUMBER format
Parse Integer	Converts a string to an INTEGER format
Checking the TIN	Checks the TIN for the checksum
Checking the existence of an attribute	Checks whether the attribute value exists
Checking values	Checks values against a regular expression. The field accepts a regular expression and a value for validation as input parameters. The value can be a string or numeric
Checking for duplicates	Checks for duplicates
Checking the link	Checks for referenced records

The DQAS should be able to establish the following types of validity indicators of attributes and characteristics of the compliance of the checked attributes with the specified rules or criteria:

1. By length – entering the boundaries that set the minimum and maximum length of the attribute.
2. By content – the degree of compliance with the subject area.

The Data Quality Model Constructor should make changes to the created Data Quality Model without restarting, stopping, or changing the program code of the DQAS.

The Data Quality Model Constructor should provide the ability to set confidence weights for the sources of this information. In addition, the data quality model should contain a single repository of quality rules that implements the data checks.

The DQAS should conduct inspections for both multiple objects and a single object. Batch checks must be initiated under the tasks created by the Admin and the Data Quality Operator. The task for conducting batch checks should include the following description of the data entity to which the quality rules should be applied. Tasks should be generated using the Data Quality Model Constructor.

The DQAS should provide the possibility of parallel execution of several tasks. The results of the checks must be saved in the Data Quality Log.

5.2 Requirements for data quality rules

If any quality control rule is violated, a table with a list of errors and information about the error must be saved in the system. For the list of errors, a data cleaning function will be provided with the ability to run on a schedule or manually.
When saving a record, the system quality rules are first checked, and then the configured quality rules of the normalization and enrichment groups are checked.

The system quality rules are permanently blocking: a record with quality errors is not saved or loaded if the verification conditions are not passed. For example, if the user tries to save a record with an empty attribute marked as required in the properties. Instead, an informational message should be displayed, and the attribute in which the error was detected is marked.

When manually entering data and saving the created or changed records in the system, non-compliance of the filled values with the quality control rules "normalization" and “enrichment.” The DQAS will issue the appropriate warnings:

1. In the upper right corner of the record card, there will be a label "Errors ()," which will display a list of errors in the quality rules.
2. The record will mark the attributes in which errors are detected.

To ensure the quality of records imported into the System, the System will provide for the formation of a list of DQ violations, which will contain records of violations of the quality control rules. In the case of a violation of the quality control rules in the entries received from external systems, the corresponding description of the violation of the rule will be recorded in the external table of the PostgreSQL database. The list of errors in the form of a report will be generated using JasperReports. Hard and soft control is performed when sending a message to the System via the SOAP API.

5.3 Use case

It is a functionally complete component of the DQAS, designed to control data quality in the Unified State Register of Real Estate, forming a data quality model, including creating the required checks and reports on the performed checks.

A diagram of the use cases for DQAS is shown in Figure 9 below.

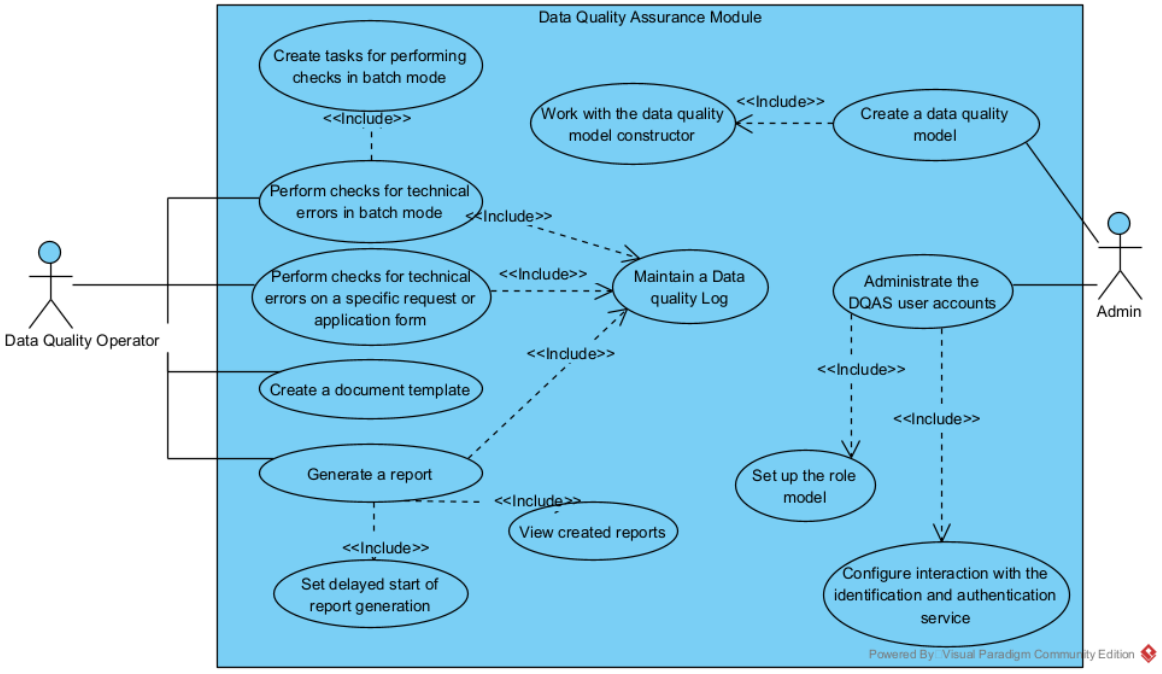

Fig. 8. The DQAS Use Case diagram.

As part of working with the application, the following processes are available to the DQAS user:

1. Create a Data Quality Model.

2. Work with the Data Quality Model constructor.
3. Check the objects of the Unified State Register of Legal Entities (including batch).
4. Generate reports.
5. Administrate the DQAS user accounts.
6. Set up the role model.

5.4 Description of TO-BE processes

As part of the examination, the team and the customer developed a methodology for reporting data in working order. The business processes identified during the examination should be recorded in this thesis using modern CASE tools designed for process modeling. When describing processes, it is allowed to use all available modern notations. The BPMN notation was used as the preferred notation for describing business processes.

The BPMN notation contains the following categories of elements:

1. Flow elements (events, processes, and gateways).
2. Connecting elements (control flows).
3. Areas of responsibility (pools and tracks) [74].

Table 6. Elements of BPMN. **[75]**

Name	Graphic BPMN symbol	Description
Start events		Show where the process starts and causes the initiation of a process. Start events cannot connect to the incoming Sequence Flow.
End Events		Indicate the end of the Sequence Flow. At the same time, other flows can continue to execute. They cannot be associated with an outgoing Sequence Flow.
Intermediate Events		Indicate where the event occurred somewhere between the start and end of the process. Affect the Sequence Flow but does not start or interrupt the process.

Table 6 (continuation). Elements of BPMN. [40]

Name	Graphic BPMN symbol	Description
The event associated with an incoming message		An incoming message triggers an event that causes the process of waiting to receive the message to continue.
The event associated with an outgoing message		An outgoing message triggers an event that causes waiting for the message to be sent to continue the process.
Exclusive Gateway OR/OR		Manages outgoing flows using Boolean expressions. Boolean expressions are based on process data.
Parallel Gateway AND		A mechanism for splitting a flow into two parallel flows or for synchronizing multiple flows into one.
Inclusive Gateway AND|OR		The expression validity is checked for each outgoing flow, and all flows for which the expression is true are activated.
Task (Standard form)		Represents an elementary action within a process. It is an indivisible work within the process. The user performs a task within the framework of the DQAS.
Sequence Flow		The arrow is used to link flow elements — events, processes, and gateways. The Sequence Flow displays the progress of the process. If necessary, the flow can be named.

5.4.1 Creating a Data Quality Model

This subsection describes the formation of the Data Quality Model: metadata of the entities of the Unified State Register of Real Estate database, their attribute composition, relations between them, and the reference books.

A diagram of the process of forming a Data Quality Model is presented below in Figure 10. The following is a basic scenario that describes the interaction in forming a data quality model.

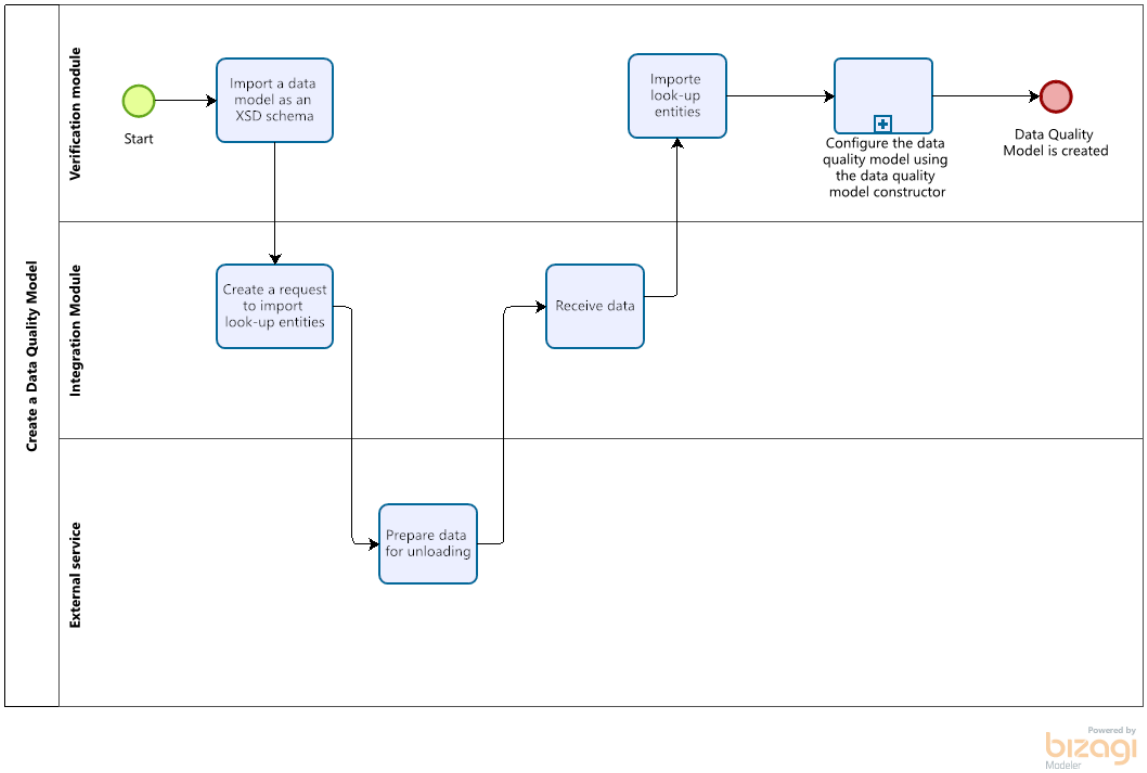

Fig. 9. Diagram of the Creating a Data Quality Model process.

The participants of the interaction are:

1. Administrator.
2. External services of the unified regulatory and reference information management system.

The criterion for success is the condition-the Data Quality Model is uploaded in the DQAS. The prerequisites for execution are: the admin user must be successfully authenticated in the system, and after logging in to the system, the admin gets to the page for creating a Data Quality Model.

The execution scenarios are described in Table 7 below.

Table 7. Execution scenarios.

Basic scenario	1. The admin uploads the data model in the form of XSD schemes into the Verification Module. 2. The integration module generates a request for the import of look-up entities to the external service. 3. The external service prepares data for unloading. 4. The integration module receives external service data. 5. The verification module imports the received look-up entities. 6. The admin configures the Data Quality Model using the Data Quality Model Constructor 7. The data quality model is created.
Alternative scenario	An alternative scenario does not exist.

5.4.2 Working with the Data Quality Model Constructor

This section describes the use case of how to work with the Data Quality Model Constructor, which includes the following subprocesses:

1. Check assignment (quality rules).
2. Creation of a new check.

A diagram of the assignation of the check process is shown below in Figure 11.

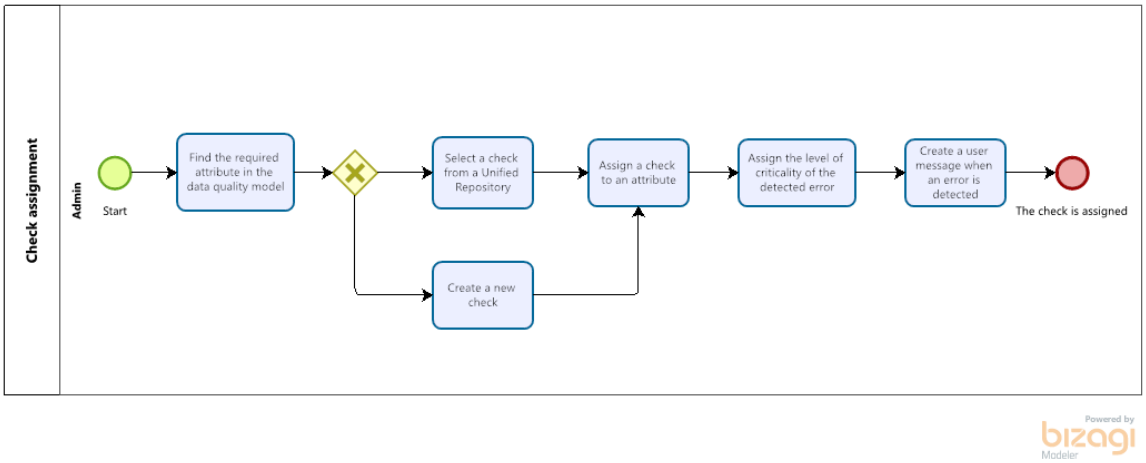

Fig. 10. Diagram of the check assignment process.

The interaction participant is the admin. The success criterion is the condition – a check is assigned to the required attribute of the object in the Data Quality Model. The prerequisites for execution are: the admin must be logged in to the DQAS, and the data model and the external service's look-up entities should be imported into the DQAS.

The execution scenarios are described in Table 8 below.

Table 8. Execution scenarios.

Basic scenario	1. The admin searches for the required attribute in the Data Quality Model. If the Unified Repository contains the required check, the admin selects it from the list. The admin assigns the check to the required attribute. The admin selects one of the four criticality levels for the detected error. The admin creates a custom message that will be displayed when an error is detected. The check is assigned.
Alternative scenario	1. If the required check is missing in the Unified Repository, the administrator creates a new check. Return to step four of the basic scenario.

5.4.3 Creating a new check

A diagram of the process of creating a new check is shown below in Figure 12.

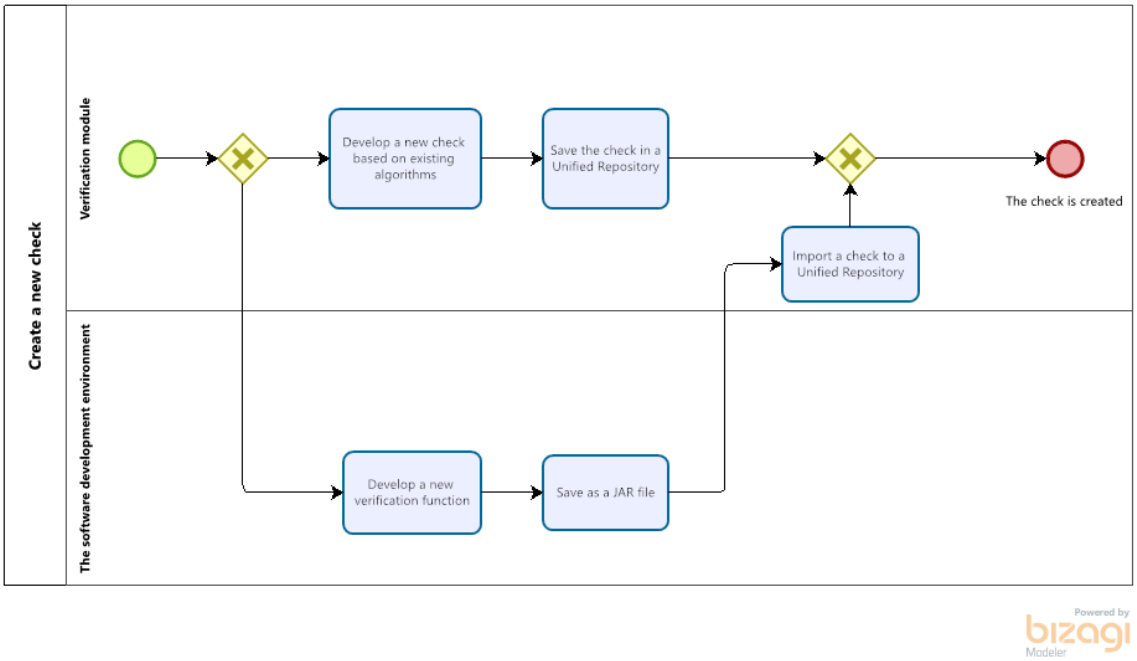

Fig. 11. Diagram of the process of creating a new check.

The following are the basic and alternative scenarios that describe the interactions while creating a new check.

The interaction participant is the admin. The criterion for success is the condition – a new check has been created in the Unified Repository (a new quality rule). The prerequisites for execution are: admin must be logged in to the DQAS, and the data model and the external service's look-up entities should be imported into the DQAS.

The execution scenarios are described in Table 9 below.

Table 9. Execution scenarios.

Basic scenario	1. If a Unified Repository contains algorithms for implementing a new check, the admin creates a new check based on them. 2. The admin saves the created check in a Unified Repository. 3. The check is created.
Alternative scenario	1. If a Unified Repository does not have the necessary algorithms for creating a new check, the function that implements such a check is created in the software development environment (not included in the DQAS). The implemented function is saved as a JAR file. Using the Data Quality Model constructor, the JAR file is imported into a Unified Repository. Return to step three of the basic scenario.

5.4.4 Perform a check

This subsection describes the process of conducting checks of one or more objects of the Unified State Register of Legal Entities, with which the Data Quality Operator works.

The scheme of the process of checking the objects is shown below in Figure 13.

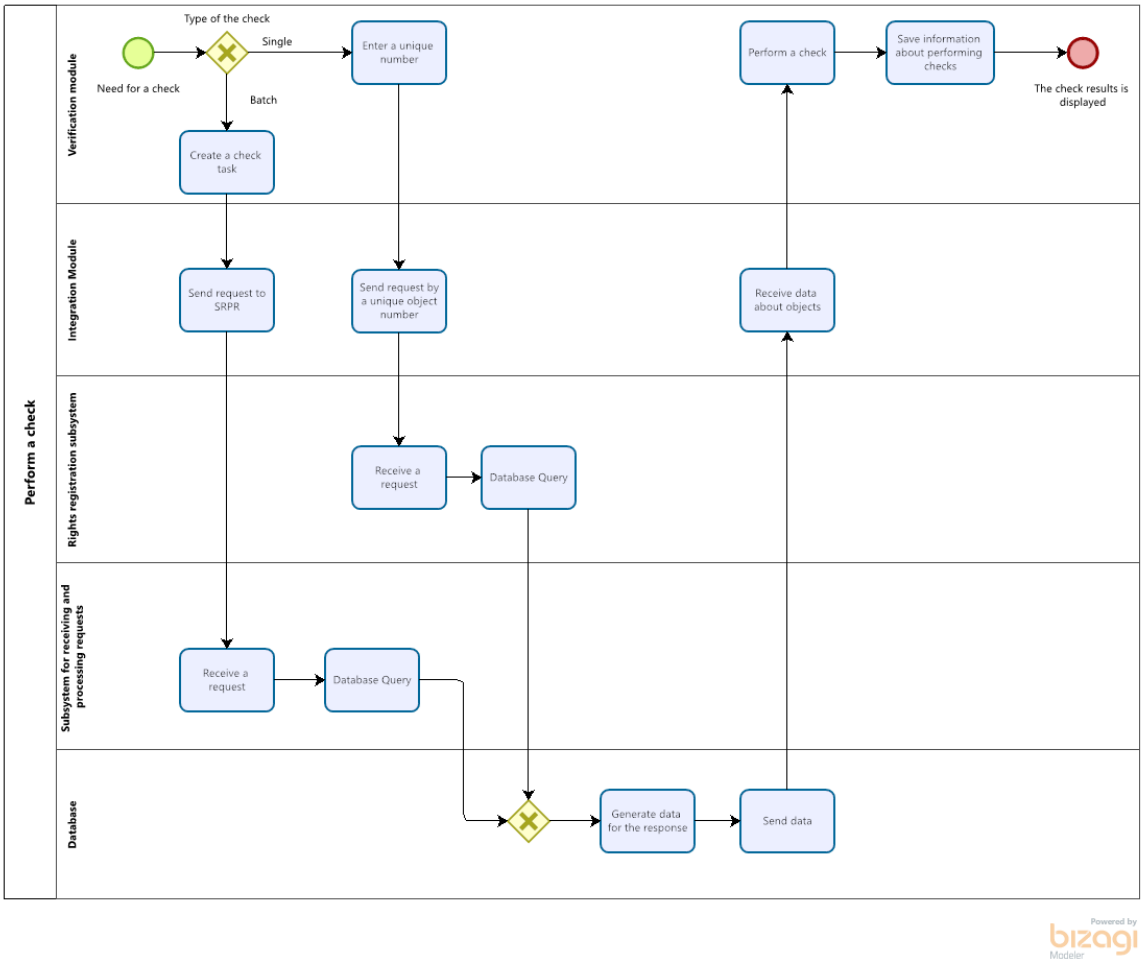

Fig. 12. The diagram of the process of checking the objects.

The following are the basic and alternative scenarios that describe the interaction of the performing checks process of the object. The Data Quality Operator is a participant in the interaction. The success criterion is when the data was checked, and the check results are displayed in the user interface. A prerequisite for execution is — The Data Quality Operator must be authorized in the DQAS.

The execution scenarios are described in Table 10 below.

Table 10. Execution scenarios.

Basic scenario	2. The Data Quality Operator needs to check the object. 3. If one object is checked, the Data Quality Operator enters the unique number of this object. 4. The integration module from the composition of the DQAS generates a request for the specified unique number to the Right registration subsystem. 5. The right registration subsystem receives a request. 6. The right registration subsystem forms a request to Database. 7. The Database generates data for response and sends it back. 8. The integration module receives data from the Database and sends it to the Verification Module. 9. Checks are carried out at the Verification module. 10. The results of the checks are saved in the Performed Log. 11. The results of the checks are shown to the Data Quality Operator
Alternative scenario	1. If it is needed to perform a mass check of objects, the Data Quality Operator creates a task to perform such a check. The task includes setting criteria for selecting objects and a list of checks that need to be performed. 2. The integration module generates a request for the specified criteria for the processing requests subsystem. 3. The processing requests subsystem receives the request. 4. The subsystem forms a request to the Database. 5. Return to step 6 of the base scenario.

5.4.5 Generating reports

This subsection describes the process of generating reports. This process is designed to process the conducted checks and present data from the Data Quality Log in visual reports.

A diagram of the report generation process objects is shown below in Figure 14.

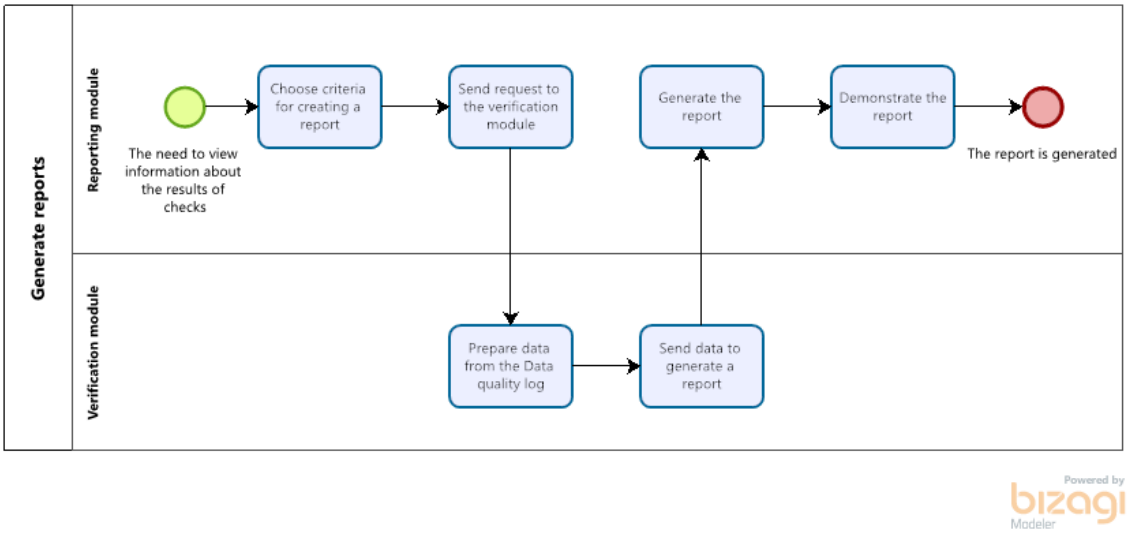

Fig. 13. Diagram of the report generation process.

The following is a basic scenario that describes the interaction in the reporting process. The Data Quality Operator is a participant in the interaction. The criterion for success is the condition – a visual report is generated under the specified criteria. The prerequisites for execution are: the data quality operator must be authorized in the DQAS, and at least one check of the objects must be performed.

The execution scenarios are described in Table 11 below.

Table 11. Execution scenarios.

Basic scenario	1. The Data Quality Operator needs to view information about the results of object checks. 2. The Data Quality Operator generates the criteria for creating the report. 3. The Reporting module generates a request to the verification module to provide data on the results of the checks. 4. The Verification module prepares the required data based on the information contained in the Data Quality Log. 5. The Verification module sends data to the Reporting Module. 6. Based on the received data, the Reporting Module generates the required report.
Alternative scenario	An alternative scenario does not exist.

5.4.6 Administration of user accounts

This subsection describes the process of managing user accounts in the DQAS. This process includes creating and modifying user accounts in the DQAS, saving all changes to the Infrastructure Service of Identification and Authentication (SIA).

A diagram of the account administration process objects is shown below in Figure 15.

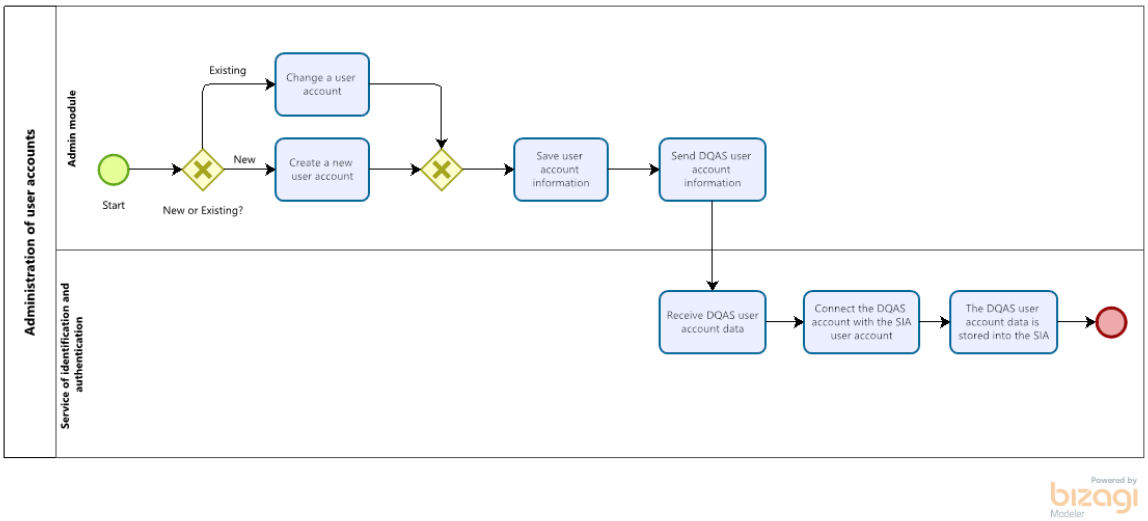

Fig. 14. Diagram of the user account administration process.

The following are basic and alternative scenarios that describe the interaction during the account administration process. The interaction participant is the admin. The data about the DQAS user account is stored in the SIA. Therefore, the user can authenticate and get access to the DQAS. A prerequisite for the implementation is that the admin must be authorized in the DQAS.

The execution scenarios are described in Table 12 below.

Table 12. Execution scenarios.

Basic scenario	1. In creating a new user account, the admin creates a combination of username and password and specifies additional information (full name, phone number) in the Administration Module. 2. The user account data is saved in the Administration Module. 3. The DQAS sends the DQAS user account data to the SIA. 4. The DQAS user account data is received in the SIA. 5. The DQAS user account is linked to the user account data in the SIA. 6. The data about the DQAS user account is stored in SIA.
Alternative scenario	1. In the case of a user account change, the admin makes the required changes (including deactivation) to the user account of the DQAS. 2. Return to step 2 of the base scenario.

The next chapter presents the implementation stages, tools, and outcomes from implementation.

6 IMPLEMENTATION STAGES AND EFFECTIVENESS

The implementation methodology is developed and is necessary for the specialists of the Implementation Department, Project Managers of the implementation, and designed to systematize the order of performance of projects. The methodology will reduce the time for planning future stages and help specialists with frequently encountered problems. All the stakeholders should make changes and additions to the method.

The participants in the development and implementation process for the project are presented in Table 13.

Table 13. Participants of the development and implementation process.

	Role name	Responsibility	Note
1	Project Manager (PM)	Communicate with the customer, monitor deadlines and resources, set tasks.	
2	Business Analyst	Conduct analytics, create a data model, develop requirements for development, check implementation, and monitor delivery times.	
3	Engineer	Install, update, and configure the platform, customization solutions, and related software.	
4	Backend Programmer	Develop modules, extensions, integration, and functions.	He is an implementation programmer.
5	Frontend Programmer	Responsible for the client-side of the user interface, the logical operation of all components of the site, including content, buttons, images, navigation, and internal links.	
6	Tester (Quality assurance)	Check and test the functionality which is developed for the project.	There may not be a person but an assigned one from the project.

Table 13 (continuation). Participants of the development and implementation process.

	Role name	**Responsibility**	**Note**
7	DevOps engineer	Work on the so-called invisible deployment when end users do not even know that a new version has been released.	
8	Technical writer	Prepare documentation.	There may not be an individual for a project.
9	Implementation Managers	Manage development and analytics departments.	Required for resource planning, checking results, and overseeing project progress.

Communication between project participants should occur: in person, by video conference, e-mail, or Skype. All project participants should be located in a familiar and unified information space, regardless of the project stage. It can be achieved by organizing general meetings, sending an e-mail to all people involved in the project, creating a general chat where vital news and results will be broadcasted. All information and outcomes are recorded in Confluence, a tool for teamwork, where accumulated knowledge is combined with opportunities for collaboration and easy access at any time and stage of the project [76].

It is necessary to create a Minutes of a Meeting (MOM) at all stages of the project, either after the meeting with the customer or internal discussions on the project. The MOM is drawn up according to a template. The MOM records the main points of discussion, such as the meeting date, the composition of participants, the list of issues, and the decision. The MOM can be either in text format or in a tabular version. The MOM should be sent to all participants of the meeting to review and consolidate the decisions taken [77]. The responsible person is PM or Analyst.

6.1 Initial stage and examination

Firstly, the formation of the goal of developing and implementing the platform (briefly, abstractly, may change and be modified later) should be made. Next, it is needed to formulate a description of the purpose. Responsible – Project Manager. The project's overall goal is displayed on the project's main page in Confluence and communicated to all project

participants at the introductory status. At the end of this stage, a top-level scope and milestones are formed from several main points that are most significant for the customer.
Then, the project manager assesses requirements for budget planning and the scope of work.

1. The project manager sends the requirements specification and other documents for evaluation.
2. An analyst is assigned to analyze the requirements and identify what needs to be finalized. It is made in the form of a list.
3. The list is sent for evaluation, which the costs are put down (in people/week or people/month). The costs should also take into account the time for testing and documenting improvements.
4. The final document is sent to the project manager.
5. It is worth paying attention to the requirements specification, which should replace the existing system since the requirements specification usually does not describe all the existing functions that users currently use.
6. The evaluation must include testing, testing individual developed features and the system as a whole, regression testing, load testing, and automation. Testing should be done on the same side as development.

After a detailed assessment of requirements, the project manager requests resources, prepares a resource plan, creates requests at Confluence and JIRA, makes platform allocation, and requests licenses. Finally, the PM determines the composition of the work and the approximate labor intensity. The labor intensity assessment should include the entire work cycle in the project and product, including testing and documentation.

The project manager organizes the first Introductory meeting of the project team, where the deadlines and tasks are announced, the order of interaction is determined, and the main issues are resolved. Then, the business analyst performs the next stage. It includes familiarization with the requirements specification and drawing up a list of tasks and improvements.

The Project Manager, Architect, and Business Analyst then conduct an examination. The examination is designed to establish interaction between the project team and the customer's team to formulate a shared vision of the project and the customer's expectations. It is

essential to understand the customer's expectations for platform performance and future functionality.

At the examination stage, the demo platform should be demonstrated for the customer, explaining how specific requirements are closed and how the platform will look so that the deployment stage will not bring any surprises. The business analyst also prepares "To Be" mockups or screenshots of the platform to minimize risks.

Besides, at the beginning of the project, the PM must agree and fix with the customer and the team how the team will deal with new requirements which are not in the requirements specification but will arise during the project (elicitation, analysis, evaluation, implementation).

The analyst and engineer deploy database dumps on our resources or accesses the databases from the customer. Then, they convert the data into the format necessary and applicable for analysis.

6.2 Design stage

The analyst determines the data sources, prepares a list and description of the data
The analyst analyses the baseline information and defines objects and relations to data sources (without attribute granularity).
As part of the analysis, the analyst must:

1. Perform data profiling using the Atacama. As a result of profiling, it should be clear: the unique values, the external keys, data formats, data statistics, duplicate data for creating Look-up entities, relations in the data, and the keys for these relations. [56]
2. Determine which objects will be Entities, Look-up entities, Classifiers, and the order and the possibility of updating them.
3. Define object attributes.
4. Identify relations.
5. Check the quality of data: inclusions of invalid characters, inclusions of Latin characters, masks.

6. Check the uniqueness of the selected foreign keys.
7. Check what is specified in the values of code attributes.

Profiling should be carried out to understand how complete the data is, the formats, and the possibility of using this data. It is also advisable to communicate with users to compare the fields (for example, database tables) and what they were working on within the interface. Furthermore, it is needed to understand how and by whom the data is used. Finally, it is necessary to check the completeness of the data: the grouping of values with a quantitative or percentage rating of each attribute; the length of the attribute.

The analyst prepares a report on the examination and use cases. After that, the project manager approves with the customer on the examination report, the description of the data model, and use cases. Next, the architect and analyst develop integration schemes (systems, exchange directions, and data flow). Then an engineer and an architect develop a design solution. Special attention is paid to the order of interaction of systems.

6.3 Development stage

An engineer installs a development platform. An analyst, an engineer, and an architect form the composition of the project distribution and determine the delivery parameters. The updated list of components that will be transferred to the customer is formed. The delivery life cycle must be agreed with the customer - the terms, scope, start and end dates of work on deliveries, testing, transmission format, installation support, and elimination of comments. Delivery time should not be too short to update the platform version, develop the tests, and document features. In addition, the team needs to check all previously developed customization.

The analyst develops a data model of the subject domain considering:

1. Division of the data by object types: entities, look-up entities, classifiers, enumerations, taking into account their features (the order of loading, the need for relations, updating).
2. Work out the types of attributes.
3. Select Code attributes for look-up entities.

4. Check the source data and the data model for empty values in the required fields.
5. Work out the relations between the objects, such as relations and relations, to look-up entities.
6. Select a foreign key for each source and object and work out the order of updating records.
7. Prepare the quality rules that should be applied at the loading stage, such as normalization and cleaning, so that the data enters the system correctly and does not require data reloading.

The analyst conducts analytics on developing external operations based on requirements and their implementation on the platform. Then the analyst makes a statement for the development of a composite function for checking the quality rules. Next, the programmer develops a composite function for checking the quality rules and tests them together with the analyst. Finally, the analyst forms the quality rules, sets up match search rules, and sets up consolidation rules.

The following should be taken into account:

1. Divide the rules into enrichment and validation
2. Work out additional custom functions that have to be developed and create tasks for their development.
3. Configure composite functions in the box solution.

After developing all the functionality and setting all the parameters at the demo stand, the platform is deployed at the customer's stand, and a team of testing engineers conducts functional and load testing.

The technological process is as follows:

1. Stand preparation.
2. Preparation of testing documentation.
3. Manual functional testing is performed. First, testing is carried out following the created checklists. After the end of this procedure, depending on the presence of defects, the following measures may be taken:
    - Functional defects, requirements defects, infrastructure defects. The test engineer should create them with a definition of criticality regarding the impact of particular functionality implementation. If the functionality is found to have

defects of priority 1 and 2, this functionality is returned for revision (the task is assigned the status “Reopen”), and the testers inform the PM;

- the task moves further through the production process with a description of the stand configuration, the stand number, and a favorable resolution.

4. Automated tests are written. If the project has a framework for automated testing and no defects in priorities 1 and 2, it is recommended to automate the new functionality. The testing engineers determine the scope and depth of automated tests with the approval of the PM.
5. Regression testing is conducted. The regression scope is compiled by the testing engineers indicating the release number. The scope is formed according to the requests of the release plan and the functionality implemented earlier. Artifacts are defects (if any) and the conclusion about the transfer of the release to the customer. A prerequisite for creating a positive conclusion is the complete passage of the scope and the absence of defects of priorities 1 and 2.

The analyst prepares The Design Solution document. The technical writer draws up documents, including the Design Solution and manuals adapted to the project. Next, the analyst and the project manager prepare the scenarios and test cases needed to demonstrate the functionality. The scenarios should be clear and logical and describe all the actions so that anyone can demonstrate the functionality on the platform. The scenario must specify exactly all the records or search criteria (in the correct case) to work with. Next, the analyst trains users to work with the platform. Finally, the analyst and engineer delete all test records, training records, and everything used during the design stages. The engineer is responsible for updating the platform and setting up the backup data. After the development and implementation are completed, the platform’s work is transferred to the technical support service. The project plan is presented in Figure 16.

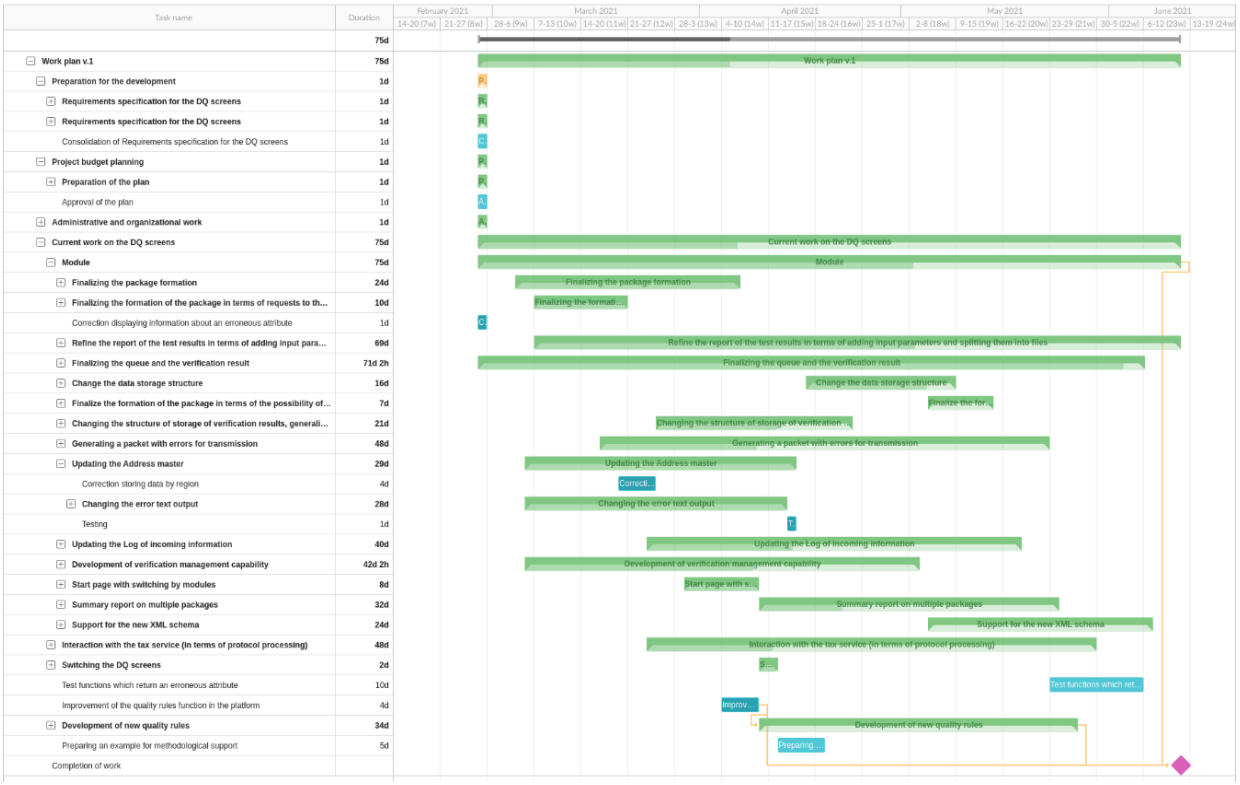

Fig. 15. Project plan of the development and implementation.

7 EVALUATION OF THE RESULTS OF THE DEVELOPMENT AND IMPLEMENTATION OF THE DQAS

In this chapter, impacts, qualitative results, and outcomes of the development are covered and discussed. Data quality management remains highly relevant to enterprises of any industry. The need to develop data quality management systems is a necessary measure in modern digital conditions, with increasing competition in the market to attract customers and ensure the effective functioning of the enterprise. It is worth noting that the development and implementation of these systems are challenging, time-consuming, and costly processes. To meet the needs of consumers, in the desire to meet the generally accepted requirements of the market, companies often spend a considerable amount of money on the development and implementation of a data quality management system. Also, many costs are needed to maintain its functioning and ensure continuous improvement.

Nevertheless, despite the difficulties of implementation and impressive costs, the competent implementation of data quality management has many absolute advantages for the company and allows it to increase the efficiency and effectiveness of the functioning of n enterprises. The absolute consequence of the implementation of DQAS in the company will be a necessary improvement in the quality of its functioning manifested in high values of the final performance indicators of the enterprise, such as effectiveness, the degree of implementation of planned activities, and achievement of planned results and performance – the ratio of the achieved results of activities to costs.

Developing an information system allows linking the planned and actual data of the enterprise to form operational reporting. Such a planning and analysis system ensures transparency of data, manageability, and flexibility in making operational management decisions. Transparency of business activities is achieved by unifying the data for various business lines and divisions and quickly and directly accessing this data for managers at various levels.

Manageability is achieved through the simultaneous use of the system by individual departments to plan, control, and manage. Although the ability to quickly adapt the data

management model to business requirements changes, external conditions provide flexibility in management.

All indicators of the economic efficiency of implementation can be divided into qualitative and quantitative. After changing how the company interacts with the system, both quantitative and qualitative indicators of the company's performance will improve. It is worth noting that the improvement of quantitative indicators results from the improvement of qualitative ones. The evaluation of the effectiveness of the implementation of DQAS is divided into two types. The economic part is a system of indicators that evaluate the economic effectiveness of the implementation. The process part assesses the compliance of the system's implementation with the enterprise's goals and objectives. In this thesis, the calculation of economic efficiency will not be covered due to the Non-Disclosure Agreement and the company's privacy.

The development and implementation's main goal was to find a solution to the problems: low speed of performing routine operations on working with data, a long decision-making time, and a high probability of technical errors.

The data quality management system achieved this goal, reducing the company's financial and time costs for working with data. With the implementation of the data quality management system, the following changes will also occur at the qualitative level:

1. A full-fledged information network was built for the well-coordinated work of departments– its creation will ensure the well-coordinated work of departments of the enterprise by creating a full-fledged information connection between them and the convenience of using one system.
2. Access rights to data were differentiated to strengthen the security of information when working with data.
3. The time for conducting legal expertise was reduced.
4. Minimized the number of unjustified decisions on suspension or refusal of the processing of requests by increasing automation of the processes of implementation of registration actions.

5. A complete and reliable database was created by improving interdepartmental information interaction with state authorities and local self-government agencies.
6. The quality of system data was improved.

The achievement of these goals was ensured by solving the following tasks:

1. Automation of the process of decision support by Registrars in the implementation of accounting and registration actions.
2. Check the information submitted to the state cadastral registration and (or) state registration of rights to identify and eliminate errors.
3. Check the information for technical errors.
4. Process, check, and enter into the information systems system received from state authorities, local self-government bodies, and other authorized agencies.

These changes are only qualitative and cannot have a quantitative representation.

Table 16 presents a comparison of the activities' critical characteristics, which allows estimating the quantitative effect of DQAS implementation. As a comparison method, the Cost-Benefit Analysis (CBA) method was chosen. It shows the DQAS implementation results by evaluating the values of critical indicators «before» and «after» implementation.

Table 14. Comparison of indicators of the analysis process.

Indicator	Before	After
ETL-process	1-2 months	1,5 – 3 h
Frequency of work with analytical applications	Once a week or once two weeks	At least 1 per day
Documentation Analysis Time	4 – 5 h (depending on the department and specialist) – on one report	30 min – 1,5 h (depending on the department and specialist) – on one report

Thus, according to the critical quantitative indicators of the development and implementation of the DQAS, the following conclusion can be drawn:

1. Reduction of the time spent by Registrar in the preparation of reporting and checking documentation by 35%.
2. Increase in labor productivity by 42%.
3. 75% increase in customer satisfaction due to fewer errors in the cadastral objects.

8 DISCUSSION AND CONCLUSIONS

The research gap thesis aimed to fill was to identify existing data quality practices and implementation of data quality assurance subsystem. Considering different aspects of data quality, this research is an excellent asset to the architectures, developers, and business analysts to develop and adopt data quality subsystems with enterprise systems.

Some issues presented in the section 2 were on a more general level and hold true for a great variety of strategic IT and business engineering initiatives. Most of the differences can be explained by the stage of the project; this case study focused on very early phase of the development, while previous studies focused on “more mature” organizations. This means that they already had established common understanding between the concepts, objectives and methods. On the other hand, our study provides ani-depth understanding of how different issues appear. Since MDM development creates major changes in an organization, ownership, accountability and responsibility issues are obviously emphasized. For example, the fear of taking on more responsibilities without receiving adequate resources is more than understandable.These concerns are largely missing from technically oriented previous studies.

Due to the prior outcomes, what are the existing practices of data quality?
In the transition to the digital economy, companies have finally become convinced that data is an important asset to properly store, process, analyze, use for making decisions, and make forecasts. The effectiveness of these processes is ensured by a single repository, into which proven quality data must be uploaded. The task of consolidating them from different sources involves comparing and synchronizing directories in different IT systems. That is why businesses need Master Data Management systems.

According to experts, data quality management solutions have become very important for the digital transformation of companies, especially those who use such emerging technologies as automation, machine learning, cloud computing and business-oriented workflows.

Data quality refers to the processes and technologies for identifying, understanding, and correcting data deficiencies that support effective decision-making and information flow management within operational business processes. Ready-to-use tools typically include important functions such as profiling and parsing text information, standardization, cleansing, matching, data replenishment, and monitoring.

What are the existing solutions of master data management systems? MDM systems are rapidly developing in the Russian market and have long been accepted in foreign marketing directions. They allow companies to focus as much as possible on the quality of data and the most complete and operational work with them.

IBS experts note the expansion of the volume of directory management, management of the main data-counterparties and materials and other necessary directories for key business processes of the enterprise. The focus is also on automating the reference information verification process, including using machine learning technologies, developing common standards for managing counterparties and materials, and creating digital ecosystems in which manufacturers and buyers can freely exchange transparent information about goods and transactions.

The defining trend remains the improvement of data quality - Data Quality. Machine learning technologies allow for better deduplication in an automated mode. In general, the development of artificial intelligence significantly changes the previously established approaches to working with reference information – it increases the efficiency of data recognition and correction, adds the ability to use multimedia information, makes data clearer.

How should the data quality assurance subsystem be developed and implemented?
The development of this direction in the enterprise allows identifying problems, shortcomings in the company's work, identifying their causes, and eliminating them urgently, without allowing customers to remain dissatisfied and dissatisfied and eventually go to competitors. Furthermore, the data is necessary for registering information in the database, which analyzes its activities to adjust its strategy. This set of tools is not limited only to the management of master data. Nowadays, software products can offer the customer

a complete software package for managing the business process. With these systems, it can be detailed the work process down to the interaction between individual employees, plus the ability to configure a unique interface for each user and distribute not only the scope of work but also the level of access to information, depending on the position and responsibilities of the user of the enterprise. This volume of functions in full allows to implement a quick solution to problems that arise in the course of work, but not only by automating the work but also by classifying the data array, with the ability to download from the database the necessary information that meets the current goals, which is helped by specially set filters.

Before implementing the MDM system, it is necessary to accurately understand the features of the enterprise, how the processes are organized in the enterprise, how effectively the interaction with suppliers and customers is carried out, how effective the exchange is in the enterprise, and the IT capabilities. In addition, it is essential to understand the system's goals and what functions are needed for implementation.

This master's dissertation involved developing the Data Quality Assurance Subsystem and a comprehensive study of the industry working with data and its quality, including MDM systems and data quality management. In the course of the work, the goal was achieved, the development of a Data Quality Assurance Subsystem was made, which allows performing operations to verify and transform data for the analysis and further work with it and automate the registrar's operations. The DQAS has been designed and developed that is ready for use for a state-owned company. Since the subsystem was developed inside the existing MDM platform, specific requirements were made for the tools used in the development since this issue remains one of the most actively discussed in the developer community to this day. During the development, agile project management methodologies were used, thanks to how the project gradually acquired new features while simultaneously passing the control of all stakeholders in the final result. The requirements for the project have been implemented – the subsystem implements all the necessary functions. Moreover, with its help, users can conduct inspections of real estate objects and speed up entering new data into the state register.

As the work progressed, the following tasks were completed:

1. The area of work with data quality, existing data quality methods, and tools were considered.

2. A literature review of systems for working with master data is presented.
3. The research method used in the thesis is disclosed.
4. The existing platform, its description, and its disadvantages are considered, and the justification for creating a subsystem is given.
5. The requirements from the customer are collected.
6. The requirements specification for the design of the future subsystem is developed according to gathered requirements.
7. The design of user cases and business processes has been completed.
8. The stages of implementation of the subsystem were given, and the project plan was described.
9. The efficiency of the implementation was given.

The developed project meets the requirements identified at the stage of setting the task. The project tasks were performed sequentially: for each required functionality, the logic of the subsystem, the service architecture was written first, and then the interface for accessing it from the client was implemented.

Thus, after the development and implementation of the DQAS, the following conclusion can be drawn:

1. Reduced time spent by Registrar in the preparation of reporting and checking documentation by 35%.
2. Increase in labor productivity by 42%.
3. 75% increase in customer satisfaction due to fewer errors in the cadastral objects.

In the future, it is planned to improve the subsystem and platform continuously, as technologies do not stand still and companies have to face many new problems that require modern solutions. In addition, at the moment, extra functions can be implemented in the subsystem that increases the attractiveness of its use. For example, an electronic signature can be added. From 2021, the approach to forming electronic signature certificates should change (in connection with the transition to the new GOST). The transition period was extended until 01.01.2022. After that, all the old certificates will become invalid. As the system uses a browser plugin for signing, it needs to be updated. Currently, the production calendar in DQAS does not have a user interface. It would be nice to add it to simplify the operation of the system. The changes will improve the system and simplify its operation for users. As future studies, it is planned to form a holistic MDM system that will monitor the performance of various organizational units of companies.

The work process in the company has become more apparent and transparent, its speed has increased significantly, and the requirements for human resources have decreased due to time reduction. Based on the results of this master dissertation, DQAS was developed and implemented at the company, employees were trained, a training system was recorded, and regulations and instructions were written. The DQAS was put into commercial operation. All the goals, objectives, and research questions in this master dissertation have been considered and fulfilled.

REFERENCES

1. P. Lepeniotis, ‘Master data management: its importance and reasons for failed implementations’, PhD, Sheffield Hallam University, 2020. doi: 10.7190/shu-thesis-00311.
2. O. Gervasi et al., Eds., Computational Science and Its Applications – ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part III, vol. 12251. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-58808-3.
3. B. Steward, ‘Writing a Literature Review’, Br. J. Occup. Ther., vol. 67, no. 11, pp. 495–500, Nov. 2004, doi: 10.1177/030802260406701105.
4. M. Pautasso, ‘Worsening file-drawer problem in the abstracts of natural, medical and social science databases’, Scientometrics, vol. 85, no. 1, pp. 193–202, Oct. 2010, doi: 10.1007/s11192-010-0233-5.
5. A. Akaev, A. Sarygulov, and V. Sokolov, ‘Digital economy: backgrounds, main drivers and new challenges’, SHS Web Conf., vol. 44, p. 00006, 2018, doi: 10.1051/shsconf/20184400006.
6. ‘Executing Data Quality Projects - 1st Edition’. https://www.elsevier.com/books/executing-data-quality-projects/mcgilvray/978-0-12-374369-5 (accessed May 12, 2021).
7. D. Loshin, Master Data Management. Morgan Kaufmann, 2010.
8. A. Al-Badi, A. Tarhini, and A. I. Khan, ‘Exploring Big Data Governance Frameworks’, Procedia Comput. Sci., vol. 141, pp. 271–277, Jan. 2018, doi: 10.1016/j.procs.2018.10.181.
9. M. Al-Ruithe, E. Benkhelifa, and K. Hameed, ‘A systematic literature review of data governance and cloud data governance’, Pers. Ubiquitous Comput., vol. 23, no. 5, pp. 839–859, Nov. 2019, doi: 10.1007/s00779-017-1104-3.
10. R. Vilminko-Heikkinen and S. Pekkola, ‘Master data management and its organizational implementation: An ethnographical study within the public sector’, J. Enterp. Inf. Manag., vol. 30, no. 3, pp. 454–475, Apr. 2017, doi: 10.1108/JEIM-07-2015-0070.

11. I. Alhassan, D. Sammon, and M. Daly, ‘Data governance activities: a comparison between scientific and practice-oriented literature’, J. Enterp. Inf. Manag., vol. 31, no. 2, pp. 300–316, Jan. 2018, doi: 10.1108/JEIM-01-2017-0007.
12. S. Traulsen, M. Tröbs, and A. Tucherpark, ‘Implementing Data Governance within a Financial Institution’, p. 15, 2011.
13. B. Otto, ‘How to design the master data architecture: Findings from a case study at Bosch’, Int. J. Inf. Manag., vol. 32, no. 4, pp. 337–346, Aug. 2012, doi: 10.1016/j.ijinfomgt.2011.11.018.
14. C. M. Olszak and E. Ziemba, ‘Critical Success Factors for Implementing Business Intelligence Systems in Small and Medium Enterprises on the Example of Upper Silesia, Poland’, Interdiscip. J. Inf. Knowl. Manag., vol. 7, pp. 129–150, 2012, doi: 10.28945/1584.
15. Microsoft SQL Server 2008 R2 Master Data Services. 2011. Accessed: May 12, 2021. [Online]. Available: https://subscription.packtpub.com/book/networking_and_servers/9781849680509
16. D. Butler and R. Stackowiak, ‘Master Data Management’, Master Data Manag., p. 61.
17. A. Dreibelbis, E. Hechler, I. Milman, M. Oberhofer, P. van Run, and D. Wolfson, Enterprise Master Data Management (Paperback): An SOA Approach to Managing Core Information. Pearson Education, 2008.
18. P. Pawluk, ‘Trusted data in IBM’s MDM: Accuracy dimension’, in Proceedings of the International Multiconference on Computer Science and Information Technology, Oct. 2010, pp. 577–584. doi: 10.1109/IMCSIT.2010.5680050.
19. C. Batini and M. Scannapieco, Data Quality: Concepts, Methodologies and Techniques. Berlin Heidelberg: Springer-Verlag, 2006. doi: 10.1007/3-540-33173-5.
20. T. N. Herzog, F. J. Scheuren, and W. E. Winkler, ‘What is Data Quality and Why Should We Care?’, in Data quality and record linkage techniques, Springer, 2007, pp. 7–15.
21. R. Y. Wang, H. B. Kon, and S. E. Madnick, ‘Data quality requirements analysis and modeling’, in Proceedings of IEEE 9th International Conference on Data Engineering, 1993, pp. 670–677.

22. H. Veregin, ‘Data quality parameters’, Geogr. Inf. Syst., vol. 1, pp. 177–189, 1999.
23. ‘THE SIX PRIMARY DIMENSIONS FOR DATA QUALITY ASSESSMENT - PDF Free Download’. https://docplayer.net/3987248-The-six-primary-dimensions-for-data-quality-assessment.html (accessed Jun. 13, 2021).
24. O. Foley and M. Helfert, ‘The development of an objective metric for the accessibility dimension of data quality’, in 2007 Innovations in Information Technologies (IIT), 2007, pp. 11–15.
25. A. R. Tupek, ‘Definition of data quality’, Census Bur. Methodol. Stand. Counc. Census Bur., vol. 6, pp. 2009–2, 2006.
26. P. Woodall, M. Oberhofer, and A. Borek, ‘A classification of data quality assessment and improvement methods’, Int. J. Inf. Qual. 16, vol. 3, no. 4, pp. 298–321, 2014.
27. H. V. Sæbø, ‘Quality assessment and improvement methods in statistics–what works?’, Statistika, vol. 94, no. 4, pp. 5–14, 2014.
28. J. Liu, J. Li, C. Liu, and Y. Chen, ‘Discover dependencies from data—a review’, IEEE Trans. Knowl. Data Eng., vol. 24, no. 2, pp. 251–264, 2010.
29. D. M. W. and J. R. Allen, ‘How artificial intelligence is transforming the world’, Brookings, Apr. 24, 2018. https://www.brookings.edu/research/how-artificial-intelligence-is-transforming-the-world/ (accessed May 04, 2021).
30. B. Otto and H. Österle, Corporate Data Quality: Prerequisite for Successful Business Models. epubli, 2015.
31. T. C. Redman, Data Driven: Profiting from Your Most Important Business Asset. Harvard Business Press, 2008.
32. C. Batini and M. Scannapieco, ‘Introduction to Information Quality’, in Data and Information Quality: Dimensions, Principles and Techniques, C. Batini and M. Scannapieco, Eds. Cham: Springer International Publishing, 2016, pp. 1–19. doi: 10.1007/978-3-319-24106-7_1.
33. R. F. Smallwood, Information Governance: Concepts, Strategies, and Best Practices. John Wiley & Sons, 2014.
34. C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, ‘Methodologies for data quality assessment and improvement’, ACM Comput. Surv., vol. 41, no. 3, pp. 1–52, Jul. 2009, doi: 10.1145/1541880.1541883.

35. L. Ehrlinger and W. Wöß, ‘A novel data quality metric for minimality’, in International Workshop on Data Quality and Trust in Big Data, 2018, pp. 1–15.
36. C. Batini and M. Scannapieco, ‘Data and information quality’, Cham Switz. Springer Int. Publ. Google Sch., vol. 43, 2016.
37. M. Ge and M. Helfert, ‘A review of information quality research—develop a research agenda’, in Paper presented at the International Conference on Information Quality 2007, 2007, pp. 76–91.
38. Y. Wand and R. Y. Wang, ‘Anchoring data quality dimensions in ontological foundations’, Commun. ACM, vol. 39, no. 11, pp. 86–95, 1996.
39. R. Y. Wang and D. M. Strong, ‘Beyond accuracy: What data quality means to data consumers’, J. Manag. Inf. Syst., vol. 12, no. 4, pp. 5–33, 1996.
40. Quality management systems: fundamentals and vocabulary. Place of publication not identified: B S I Standards, 2005.
41. T. N. Herzog, F. J. Scheuren, and W. E. Winkler, Data Quality and Record Linkage Techniques. Springer Science & Business Media, 2007.
42. ‘Managing data quality to optimize value extraction | Delta Partners Group’. https://www.deltapartnersgroup.com/managing-data-quality-optimize-value-extraction (accessed May 04, 2021).
43. R. Y. Wang, M. P. Reddy, and H. B. Kon, ‘Toward quality data: An attribute-based approach’, Decis. Support Syst., vol. 13, no. 3–4, pp. 349–372, Mar. 1995, doi: 10.1016/0167-9236(93)E0050-N.
44. R. VasanthKumarMehta and S. Rajalakshmi, ‘Semantic Integrity Constraint Rule Discovery and Outlier Detection in Relational Data as a Data Quality Mining Technique’, Int. J. Comput. Appl., vol. 88, no. 6, pp. 23–26, Feb. 2014, doi: 10.5120/15357-3819.
45. G. Sivathanu, C. P. Wright, and E. Zadok, ‘Ensuring data integrity in storage: techniques and applications’, in Proceedings of the 2005 ACM workshop on Storage security and survivability, New York, NY, USA, Nov. 2005, pp. 26–36. doi: 10.1145/1103780.1103784.
46. D. S. Tahat and K. Ahmad, ‘Semi-Automated Schema Integration (Icase): A Tool To Identify And Resolve Naming Conflicts’, 2013.

47. K. P. Kusuma Dewi, T. Fabrianti Kusumasari, and R. Andreswari, ‘Analysis and Design of Architecture Master Data Management (MDM) Tools for Open Source Platform at PT XYZ’, in 2019 5th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, Jul. 2019, pp. 1–6. doi: 10.1109/ICST47872.2019.9166255.
48. H. Galhardas, D. Florescu, D. Shasha, and E. Simon, ‘Declaratively cleaning your data using AJAX’, 2000.
49. H. Galhardas, D. Florescu, D. Shasha, and E. Simon, ‘AJAX: an extensible data cleaning tool’, ACM SIGMOD Rec., vol. 29, no. 2, p. 590, May 2000, doi: 10.1145/335191.336568.
50. M. A. Hernández and S. J. Stolfo, ‘Real-world data is dirty: Data cleansing and the merge/purge problem’, Data Min. Knowl. Discov., vol. 2, no. 1, pp. 9–37, 1998.
51. M. L. Lee, H. Lu, T. W. Ling, and Y. T. Ko, ‘Cleansing data for mining and warehousing’, in International Conference on Database and Expert Systems Applications, 1999, pp. 751–760.
52. A. E. Monge, ‘Matching algorithms within a duplicate detection system’, IEEE Data Eng Bull, vol. 23, no. 4, pp. 14–20, 2000.
53. Y. Yang, S. J. Adelstein, and A. I. Kassis, ‘Target discovery from data mining approaches’, Drug Discov. Today, vol. 17, pp. S16–S23, 2012.
54. S. J. Wilson, ‘Data representation for time series data mining: time domain approaches’, Wiley Interdiscip. Rev. Comput. Stat., vol. 9, no. 1, p. e1392, 2017.
55. A. Gal, H. Roitman, and R. Shraga, ‘Heterogeneous data integration by learning to rerank schema matches’, in 2018 IEEE International Conference on Data Mining (ICDM), 2018, pp. 959–964.
56. V. Raman and J. M. Hellerstein, Potter’s Wheel: An Interactive Framework for Data Cleaning and Transformation.
57. P. Xue et al., ‘Fault detection and operation optimization in district heating substations based on data mining techniques’, Appl. Energy, vol. 205, pp. 926–940, 2017.
58. R. Mukherjee and P. Kar, ‘A comparative review of data warehousing ETL tools with new trends and industry insight’, in 2017 IEEE 7th International Advance Computing Conference (IACC), 2017, pp. 943–948.

59. B. E. Elbaghazaoui, M. Amnai, and A. Semmouri, ‘Data Profiling over Big Data Area’, in Intelligent Systems in Big Data, Semantic Web and Machine Learning, N. Gherabi and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2021, pp. 111–123. doi: 10.1007/978-3-030-72588-4_8.
60. Entity Information Life Cycle for Big Data. Elsevier, 2015. doi: 10.1016/C2013-0-18748-X.
61. ‘Data Standardization - an overview | ScienceDirect Topics’. https://www.sciencedirect.com/topics/computer-science/data-standardization (accessed May 12, 2021).
62. D. V. No et al., Data Engineering.
63. M. G. Kahn et al., ‘A Harmonized Data Quality Assessment Terminology and Framework for the Secondary Use of Electronic Health Record Data’, eGEMs, vol. 4, no. 1, Sep. 2016, doi: 10.13063/2327-9214.1244.
64. D. International, DAMA-DMBOK: Data Management Body of Knowledge: 2nd Edition, Second edition. Basking Ridge, New Jersey: Technics Publications, 2017.
65. K. M. Eisenhardt and M. E. Graebner, ‘Theory Building From Cases: Opportunities And Challenges’, Acad. Manage. J., vol. 50, no. 1, pp. 25–32, Feb. 2007, doi: 10.5465/amj.2007.24160888.
66. W. J. Orlikowski and J. J. Baroudi, ‘Studying Information Technology in Organizations: Research Approaches and Assumptions’, Inf. Syst. Res., p. 29, 1991.
67. A. L. Strauss and J. M. Corbin, Basics of qualitative research: techniques and procedures for developing grounded theory, 2nd ed. Thousand Oaks: Sage Publications, 1998.
68. ‘Research Design: Qualitative, Quantitative, and Mixed Methods Approaches - John W. Creswell - Google Книги’. https://books.google.ru/books/about/Research_Design.html?id=4uB76IC_pOQC&redir_esc=y (accessed May 12, 2021).
69. T. C. Lethbridge, S. E. Sim, and J. Singer, ‘Studying Software Engineers: Data Collection Techniques for Software Field Studies’, Empir. Softw. Eng., vol. 10, no. 3, pp. 311–341, Jul. 2005, doi: 10.1007/s10664-005-1290-x.

70. B. B. Kawulich, ‘Participant Observation as a Data Collection Method’, Forum Qual. Sozialforschung Forum Qual. Soc. Res., vol. 6, no. 2, Art. no. 2, May 2005, doi: 10.17169/fqs-6.2.466.
71. G. A. Bowen, ‘Document Analysis as a Qualitative Research Method’, Qual. Res. J., vol. 9, no. 2, pp. 27–40, Jan. 2009, doi: 10.3316/QRJ0902027.
72. ‘Demographics & Methodology 2019 - The state of Developer Ecosystem in 2019 Infographic’, JetBrains. https://www.jetbrains.com/lp/devecosystem-2019/demographics/ (accessed Apr. 28, 2021).
73. K. Schwaber, Agile Project Management with Scrum. Microsoft Press, 2004.
74. Silver, Bruce: BPMN Method and Style, 2nd Edition, with BPMN Implementer’s Guide: A structured approach for business process modelling and implementation using BPMN 2.0. Cody-Cassidy Press, 2011.
75. D. Jones and R. Brown, ‘Business Process Management’, p. 10, 1851.
76. S. Kohler, Atlassian Confluence 5 Essentials. Olton, UNITED KINGDOM: Packt Publishing, Limited, 2013. Accessed: Apr. 26, 2021. [Online]. Available: http://ebookcentral.proquest.com/lib/unilu-ebooks/detail.action?docID=1192657
77. J. Gutmann, Taking Minutes of Meetings. Kogan Page Publishers, 2016.