Quality Management in the Public Sector: Definition and Prioritization of Indicators for Assessment and Decision-Making

Valéria Lelli*
Computer Science Department,
Federal University of Ceará (UFC)
Fortaleza, Ceará, Brazil
valerialelli@ufc.br

Fabiana Gomes Marinho*
Federal Institute of Ceará (IFCE)
Fortaleza, Ceará, Brazil
fabiana.gomes@ifce.edu.br

Lucas Cabral Amador Mourão
Federal University of Ceará (UFC)
Fortaleza, Ceará, Brazil
lucasmourao@alu.ufc.br

Paulo Afonso Pamplona
Federal Institute of Ceará (IFCE)
Fortaleza, Ceará, Brazil
junior.pamplona07@aluno.ifce.edu.br

Lucas Ferreira de Almeida
Federal Institute of Ceará (IFCE)
Fortaleza, Ceará, Brazil
lucas.ferreira.almeida62@aluno.ifce.edu.br

Rossana M. de Castro Andrade
Computer Science Department,
Federal University of Ceará (UFC)
Fortaleza, Ceará, Brazil
rossana@ufc.br

Ismayle de Sousa Santos
State University of Ceará (UECE)
Fortaleza, Ceará, Brazil
ismayle.santos@uece.br

Juliana Lopes Gurgel
julianalgurgel@alu.ufc.br
Federal University of Ceará (UFC)
Fortaleza, Ceará, Brazil

Maria Inês Vale Silva
Federal University of Ceará (UFC)
Fortaleza, Ceará, Brazil
ines.vale@sefaz.ce.gov.br

ABSTRACT

The use of historical data in software projects, when aligned with quality goals and business objectives, has increasingly been recognized as a strategic approach to enhancing both management and system development. This practice relies on the analysis of collected data, which enables the identification of opportunities and the anticipation of risks. As quality processes are implemented, these data are transformed into measures and metrics used to develop indicators, allowing for continuous performance monitoring and driving organizational improvement. In this context, this report presents the process used for the definition, validation, and prioritization of quality measures and indicators applied to the management and development of software projects in a public organization. Furthermore, in this study we discuss the main challenges faced, as well as the lessons learned and the benefits observed at the current stage of the process of implementing this practice within the institution.

KEYWORDS

Software Quality Management, Software Process, Quality Indicators and Measures

1 Introduction

The use of historical data in software projects serves multiple purposes: it informs planning and estimation of activity deadlines, supports activity management and monitoring, aligns quality and business goals, and drives improvements in communication, processes, and development tools [14][30].

Data-driven practices have enhanced evidence-based decision-making [38]. In addition, the systematic collection of measures, the definition of metrics, and the development of indicators support planning, estimation, monitoring, and risk mitigation [16][20]. However, converting unprocessed data into meaningful indicators continues to face recurring challenges, including data quality and availability issues, as well as integration with existing tools and workflows [37]. Furthermore, cultural barriers and limited analytical maturity hinder the effective adoption of data-driven practices [25].

To address these challenges, recent studies have focused on defining measures or indicators to assess organizational performance at different levels. For example, Junior et al. [22] propose a structured process to build organizational performance indicators in the public sector, while Carvalho et al. [4] present a set of 39 software requirements measures applied in an industrial software project using Goal-Question-Metric (GQM) framework [2][36]. These studies demonstrate how structured approaches to defining indicators can support more informed decision-making, both in public administration and in software development contexts.

In this paper, we present a structured process for defining, validating, and prioritizing quality indicators developed through a research project conducted in collaboration with a public institution. The project integrated established best practices in quality management with theoretical foundations drawn from the specialized literature on software metrics, measurement, and analysis.

The proposed process consisted of four main sub-processes: Definition, Validation, Implementation, and Prioritization of Indicators. This process was developed based on a literature review and an analysis of relevant ISO (International Organization for Standardization) standards [17][18][19]. Additionally, we utilized our experience from two similar Research, Development, and Innovation (R&D&I) projects conducted in partnership with the software industry [4] and the public sector [7][39]. Unlike [4], our process employs the Data Canvas strategy [33] to systematically identify stakeholders' needs. From the execution of this process, we derived a total of 159

*Both authors contributed equally to this research.

quality indicators based on 209 measures and identified eight key lessons learned.

The remainder of this paper is organized as follows. Section 2 introduces the key concepts related to process quality in software development. Section 3 presents related work concerning the indicators in software projects. Section 4 outlines the proposed process for defining quality indicators to support the monitoring of software management and development processes. Section 5 presents the main findings of the study and Section 6 presents the lessons learned, and, finally, Section 7 presents the threats to validity, and Section 8 presents the final considerations.

2 Theoretical Background

The theoretical background is divided into four main sections focused on software process quality, covering key concepts of software quality, the definitions of measures, metrics, and indicators, the software measurement process, and the methodologies used in designing and implementing the proposed process.

2.1 Measures, Metrics, and Indicators

This subsection outlines the primary concepts that base the quantitative assessment of software quality: measures, metrics, and indicators. A clear distinction among these concepts is essential for establishing a systematic approach to monitoring and improving both software products and development processes.

A measure is a quantitative value obtained directly from an attribute of a software product, process, or resource, following predefined rules with the goal of reducing uncertainty and minimizing subjectivity in software evaluation [11][20].

The measurement process begins with data collection, which may be carried out through automated tools or manual inspections. The collected data is then recorded in an organized and structured manner, ensuring traceability between the analyzed artifacts and the corresponding measurement values.

Finally, the collected data are validated to ensure its consistency and accuracy, thereby confirming that it reliably reflects the target attribute [28].

A metric is a function or a relationship between two or more measures. Metrics transform raw measures into interpretable information, enabling analysis and comparison across time, processes, or projects [28][34]. They typically consist of a formula or procedure that converts collected data into standardized numerical values [11]. To ensure consistency and reliability, metrics must be clearly defined to avoid ambiguities in data collection [24]. The effectiveness of process improvement initiatives relies on the availability and accuracy of metrics captured throughout the software development lifecycle.

An indicator consists of a metric (or set of metrics) interpreted within a context to support management decisions and continuous improvement.

While metrics provide quantitative data, indicators translate this data into contextualized information, enabling performance assessment, trend detection and decision-making [16][20].

2.2 Software Measurement Process

The measurement process involves the collection and analysis of data to generate indicators used for monitoring and evaluating software attributes and their associated processes. It begins with the establishment of measurement objectives, which clearly define the aspects of the product or process to be assessed, ensuring alignment with the organization's strategic goals [20].

After that, data collection is initiated and must be conducted in a consistent, reliable, and well-documented manner, using appropriate methods to ensure the quality and integrity of the information obtained [28]. The raw data is then processed and transformed into relevant information, supporting the construction of quantitative indicators that base assessment and inform decision-making [3].

Quality indicators play a fundamental role in supporting strategic management and the continuous improvement of processes, products, and projects within the context of software development. Their use enables the systematic monitoring of activity progress, the objective evaluation of deliverable quality, and the proactive identification of opportunities for improvement [28].

Once specific attributes — such as effort, productivity, defect density, delivery cycle time, and customer satisfaction — are quantified, indicators convert subjective perceptions into concrete, comparable data that support decision-making across various organizational levels [3][34].

Using indicators systematically selected, as recommended by standards such as ISO/IEC 15939:2017 (Systems and Software Engineering — Measurement Process) [20] and maturity models like CMMI (Capability Maturity Model Integration) [6], enables development teams to align operational practices with the organization's strategic goals, as well as enhance planning, and reinforce quality control and performance management mechanisms.

In this sense, indicators serve as essential instruments for promoting transparency, guiding corrective actions, and driving the continuous improvement of software processes.

2.3 Data Canvas

The Data Canvas is a strategic planning instrument designed to support the creation and development of business models, whether new or already established [26]. Within the context of project management, it plays a base role in organizing and aligning data usage in a visual and collaborative manner, enabling its structured collection and application, and fostering a culture of evidence-based decision-making [29]. By mapping data flows in alignment with project goals, the Data Canvas also functions as a communication tool among the departments involved, promoting shared understanding and minimizing misalignments [8][10].

The Data Canvas is particularly valuable in collaborative settings such as workshops, project kickoff meetings, and review sessions, as it ensures that business, IT, and analytics teams develop a shared understanding of their respective roles within the data flow. By linking business questions to performance indicators and analytical hypotheses, the tool enables the identification of relevant measures while eliminating low-impact or redundant indicators [5]. Furthermore, it supports the mapping of the entire data lifecycle, by defining data sources, update frequency, and role-based responsibilities.

Among the key questions that the Data Canvas helps to address are: What data is required to answer the business questions? Which data sources are available? How will the data be transformed into value? Who holds responsibility at each stage of the data management process?

In summary, the Data Canvas is more than a technical tool, but rather a strategic instrument for ensuring that projects are data-driven in an efficient, ethical, and goal-oriented manner [9]. By structuring information into nine interrelated blocks that describe the business model and its operational dynamics, the tool facilitates the transformation of data into informed decisions and, ultimately, into organizational value.

3 Related Work

The literature on quality management in software projects, particularly those focusing on indicators, emphasizes three critical aspects to consider: (i) defining the purpose of measurement; (ii) ensuring the quality of data sources; and (iii) documenting the intended uses for each indicator [1][13]. However, literature reviews show that many proposals remain generic and fail to explain how to operationalize data collection and consistent usage in real-world contexts [1] [35].

Several studies have focused on defining indicators or measures addressing different topics within the software context (*e.g.,* Requirements, Design and Implementation [27]). Others concentrate on a specific topic (*e.g.,* Requirements [4]) or on organizational-level aspects [22].

For example, Carvalho et al. [4] propose 39 measures specific for *Requirements* through the use of the GQM approach in an industrial software project. In a broader context, Poornima and Suma [27] classify metrics across Requirements, Design, and Implementation phases, highlighting the link between process and product, but they omit Management, Testing, and Security and do not validate the indicators in practice.

Regarding the definition of indicators, the processes used to define them vary, ranging from empirical approaches Rashid et al. [31] to stakeholder-driven techniques such as Canvas and checklists Helenius [15], or to the use of other artifacts such as guides, official documents, and manuals Junior et al. [22]. While Junior et al. [22] present a structured process for constructing performance indicators, consisting of four phases and 14 steps and serving as a practical reference for both researchers and managers, their study focuses on performance indicators in public administration rather than software contexts. By comparison, Helenius [15] address institutional contexts within software-related projects, proposing metrics aligned with critical success factors using the Project Canvas [12]. Although the study focuses on measures of overall success, our proposal categorizes indicators by operational domain. Similarly, Rashid et al. [31] provide a catalog but do not outline any process for selection or customization. To address this gap, our process incorporates team-based validation and defining tailored measures.

Junior et al. [22] introduced a structured framework for designing performance indicators in public administration. Their study conducted a qualitative analysis of 39 official documents, including government guides and manuals. The authors identified 14 methodological approaches, 43 attributes, and 20 validation criteria, structured into four phases and 14 stages for building, assessing, and validating performance indicators. The framework tackles challenges such as lack of standardization and unreliable data, offering a systematic method that strengthens transparency, accountability, and evidence-based decision-making in public management.

Most of the studies mentioned above also lack validation in real-world settings, particularly with respect to the implementation and practical application of the proposed indicators. To address some of these challenges, the literature recommends using practices such as workshops, canvases, and checklists to map data sources and responsibilities [13][23]. It additionally advises documenting each indicator's formula, frequency, and origin, and testing them in real scenarios to ensure they are interpretable and widely accepted [13][17][18][19][32]. Nevertheless, despite these recommendations, detailed definitions and documentation of measures, metrics, and indicators are still scarce.

In this regard, and to address the identified gaps in validation, documentation, and practical applicability, this study proposes a process for defining, validating, and prioritizing indicators tailored to institutional contexts.

4 Quality Indicators Definition Process

This section outlines the process of defining quality measures, metrics, and indicators used to monitor software management and development processes. As illustrated in Figure 1, the activities' process involve four main teams in the early stages: **Quality and Processes**, responsible for defining quality indicators aligned with the client's needs and implementing the processes required for data collection; **Data Science**, responsible for understanding the problem, collecting data, modeling Machine Learning algorithms, evaluating models, and generating reports; **Requirements**, responsible for technical decisions and processes related to project requirements; and **Client**, responsible for defining the project goals, providing domain knowledge and business context, making data available and ensuring access to information sources.

4.1 Consulting Analysis

The first activity consisted of the **Analysis from Prior Consulting** services in process management and quality, previously contracted by the public institution, which had proposed a restructuring of internal IT process management. This activity involved reviewing the artifacts produced by the consultancy to identify the results achieved and interpret the proposed action plan.

Initially, the defined processes were categorized into *Requirements Management*, *Solution Identification and Development*, and *Change Acceptance and Transition*, and subsequently analyzed. However, the level of detail available in the supporting materials was limited, which constrained the scope of the analysis and hindered a more in-depth understanding of these processes. Subsequently, the indicators were specified based on the control criteria established by the COBIT 5 framework [21].

Finally, the action plan aimed at improving the internal customer experience, increasing the effectiveness of solution delivery by the IT department, and mitigating the gaps identified in the mapped processes was evaluated. The identified gaps revealed deficiencies related to project management, highlighting the need

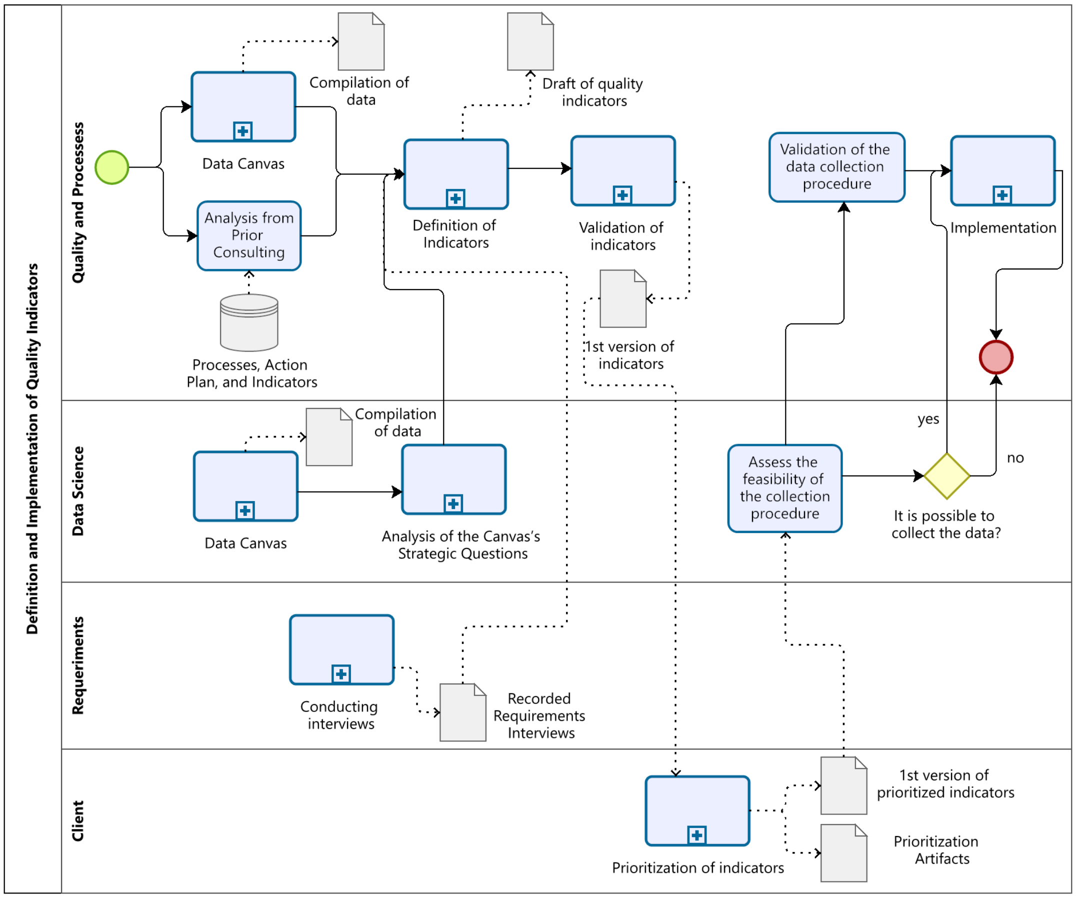

Figure 1: Process for defining the quality indicators

for investment in improving and defining specific processes. This diagnosis reinforced the importance of reviewing existing practices and adopting approaches aligned with organizational needs.

4.2 Data Canvas

In parallel with the **Analysis from Prior Consulting**, a process was conducted to identify, analyze, and refine the set of measures, metrics, and indicators for software development and quality management. This process was carried out through a collaborative workshop that brought together key company stakeholders (managers, developers, and business analysts) around a **Data Canvas**. The workshop included representatives from various internal organizational units, such as *IT Governance and Security*, *IT Projects and Consulting*, *Systems Development*, *IT Infrastructure Management*, *Data Intelligence*, as well as business areas.

The session began with a brief presentation explaining the purpose and structure of the **Data Canvas** Workshop. Participants were then divided into groups composed of members from six different organizational units. Each group was responsible for discussing and completing the canvas based on the following sections:

- **Objectives/Outcomes** the organization aims to achieve in the short and medium term. For example, "increase application security" and/or "reduce the number of errors";
- **Strategic Questions/Decisions** the organization should ask to better understand and qualify the objectives or outcomes;
- **Required Data** needed to answer the strategic questions;
- **Data Products** that help answer the strategic questions based on the required data;
- **Internal Engagement** to define actions and behaviors that promote internal involvement and support decision-making;
- **External Engagement** to define actions and behaviors that foster external involvement and build data-driven narratives.

At this initial stage, there was an open discussion about which business decisions needed to be supported by evidence, which information sources were available in internal systems (such as code repositories, backlog tools, and defect databases), and which indicators offered the greatest potential to measure key aspects of the project, such as process efficiency, product quality, and customer satisfaction.

The **Data Canvas** workshop lasted two and a half hours and included 23 participants, 17 of whom represented the client, while the remaining six facilitated the execution of the workshop. A total of four groups were formed, each consisting of approximately 5 to 6 participants.

During the workshop, the groups received support from experts, such as Data Scientists and researchers who guided the discussions and clarified questions regarding the completion of the canvas. After completing the canvas, a representative from each group presented the results of their work. Each presentation was recorded for later use during the data compilation phase.

At the end of the workshop, four Data Canvases were produced. Figure 2 illustrates an excerpt from a completed canvas. These canvases were then analyzed by the **Quality and Process** and **Data Science** teams to review the strategic questions and assess the feasibility of defining indicators and data products that would meet the needs of the client's internal organizational units.

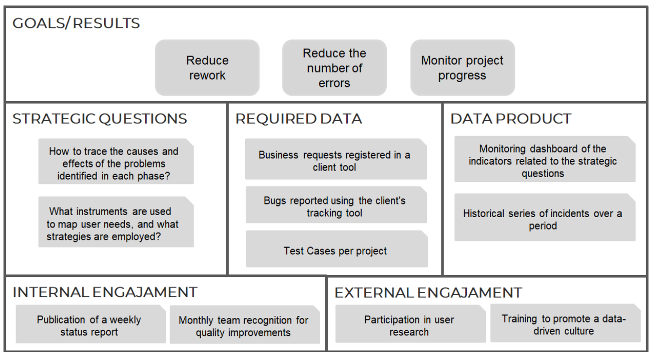

Figure 2: Sample Data Canvas for Project Planning

4.3 Definition of Quality Indicators

Figure 3 presents the activities related to the **Definition of Indicators** subprocess.

To define quality indicators, the activity **Analysis of Similar Projects** was carried out. We leveraged our previous experience in two similar Research, Development, and Innovation (R&D&I) projects [4][7][39] in collaboration with the software industry.

In the first project, we aimed to improve the client processes, focusing on the areas of Requirements and Testing. Thus, we defined 39 requirements [4] and 41 testing measures using the Goal-Question-Metric (GQM) approach. In the second project, we developed a platform to support the decision-making of public managers at a city hall [7][39]. The platform requirements were based on challenges previously identified by a Data Canvas Workshop. Based on the canvas analysis, strategic questions were identified, and data scientists defined the data products and outlined the process for acquiring, processing, and analyzing the data. The goal was to analyze the data registered on the platform to answer the challenge's strategic questions.

We analyzed both the set of requirements and testing measures from the first project and the data-driven approach used to address strategic questions in the second project to provide more relevant indicators for the current project.

Next, a technical and scientific **Literature Review** was conducted through bibliographic research in scientific databases such as IEEE Xplore, Scopus, and the ACM Digital Library, aiming to identify established practices and recent innovations related to indicators. The review enabled the mapping of measures and indicators applicable to the areas of *Project Management*, *Requirements*, *Testing*, *Development*, and *Security*.

Complementing this approach, the ISO/IEC standards 25022 [18], 25023 [19], and 25024 [17] were examined, detailing definitions of quality in use, product quality, and data quality, respectively. Upon reviewing each standard, standardized measures, formulas, and quality attributes were identified and classified according to effectiveness, efficiency, reliability, and other relevant dimensions. A systematic comparison between these normative metrics, the indicators proposed by the external consultancy, and those identified in the literature ensured a comprehensive portfolio aligned with international best practices.

An **Analysis of the flows in the client's systems** was also conducted to map the sequence of processes followed in the projects of the contracting institution. The institution utilizes various issue tracking tools; therefore, we analyzed the client's tool workflows as well as their methods of collection to identify relevant measures and indicators. The tools Assyst[1], ClickUp[2], and Azure Boards[3] were analyzed.

Initially, every client service request must start with a request opened in the *Assyst* tool. This request can be created either by a manager or a member of the technical team. Requests can be categorized into the following types: error/failure, enhancement, unavailability, and other types of requests. The lifecycle of a request in *Assyst* includes the following status: open, pending, resolved, closed, and canceled. Additionally, each request must include a resolution timeframe, which the technical team may adjust as needed. Once a request is resolved, it is closed; if it needs to be reopened, a new request must be created in *Assyst*. All requests have a history log that captures the status transitions throughout their lifecycle, along with other details such as the date, time, and internal organizational unit of the client responsible for submitting the request.

Therefore, a request opened via *Assyst* is analyzed by the technical team, and if it is valid, the team incorporates it as an activity to be added to the Sprint for the target system. This request can generate one or more tasks in ClickUp or Azure Boards, or mapped to any other tool used to track a Sprint's backlog. In the case of *ClickUp*, it is customized by workspaces. These can be organized by application type (*e.g.,* mobile) or by a specific internal organization of the client, or even by business areas of the systems developed by the client. Thus, the request starts via *Assyst*, generating one or more tasks via *ClickUp*. A task that must progress from creation to completion, following the statuses: "Open", "In Progress", "In Testing", "In User Acceptance Testing (UAT)", "Pending", "Canceled", "Approved", and "Completed", representing the entire life cycle up to production delivery. When a task is completed via *ClickUp*, it must also be completed via *Assyst*.

Azure Boards is another tool that can be used for Sprint tracking. The Analysis of Azure Boards was conducted with a focus on its core components (Boards, Backlogs, and Sprints) to understand the life cycle management of activities and identify data collection points for project indicators. Initially, the hierarchical

[1] https://www.assyst.net/
[2] https://clickup.com/
[3] https://azure.microsoft.com/

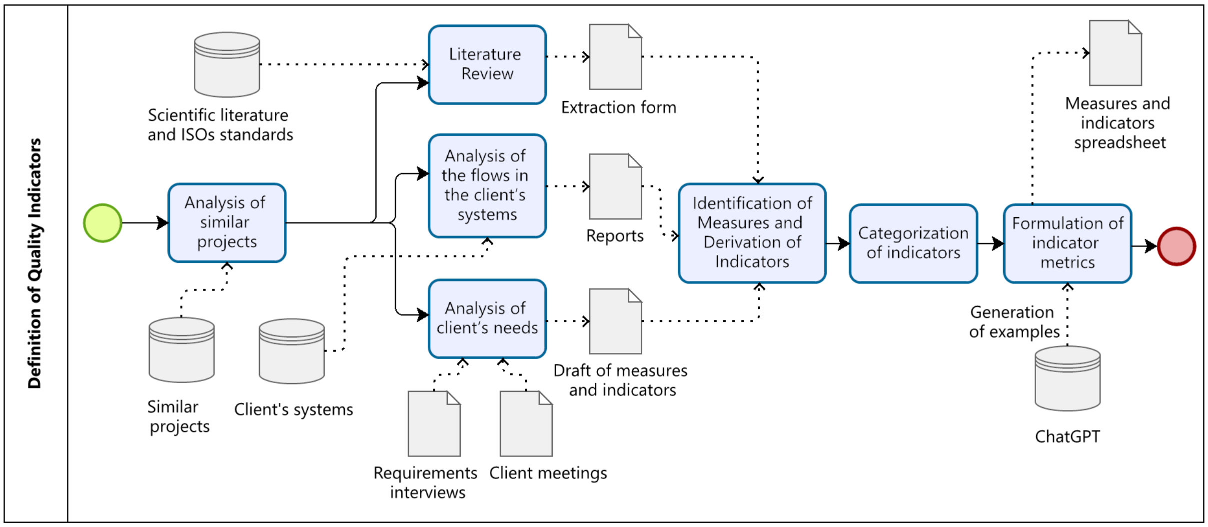

Figure 3: Subprocess for defining quality indicators

organization of work items was examined. The tool employs distinct Work Item Types (such as Epics, Features, User Stories, and Bugs), enabling a parent-child relationship that facilitates aggregated progress tracking. The workflow is defined by the transition of these items through customizable states that represent the team's process, such as New, Active, Resolved, and Closed, serving as the foundation for measuring the flow.

The Analysis of the client's tools enabled the precise mapping of data collection points for the previously defined indicators, aligned with the organization's strategic goals, as well as the identification of opportunities for creating new measures based on the organization's workflow. The study provided a comprehensive and continuous view of activity progress within client tools, allowing for full tracking of the life cycle—from task initiation to closure—and the derivation of critical metrics such as cycle time, rejection rate, volume of rework, and team productivity.

In parallel, the **Requirements** team conducted interviews with leaders from various departments of the public institution with the goal of performing an **Analysis of the client's needs**. These interviews aimed to gather insights regarding expectations, suggestions for indicators to be analyzed, challenges identified by participants in implementing improvements, and aspects related to monitoring the defined indicators. All interviews were recorded and transcribed. This material was then analyzed by the **Quality and Processes** team, focusing on identifying patterns, recurring themes, and relevant information expressed by the interviewees.

Based on this analysis, a set of strategic measures and indicators was defined, designed to incorporate the suggestions provided, reflect the expectations of the different departments involved, and address the challenges related to implementing the proposed improvements. Tables 1 and 2 present the recurring themes and challenges identified during the interviews with department leaders.

In the activity **Identification of Measures and Derivation of Indicators**, we analyzed the artifacts by considering findings from

Table 1: Leaders' needs

Recurring themes
Team productivity visibility
Institutionalization of diverse testing activities
Monitoring of team schedules
Identification of training needs
Client value perception
Control of the homologation and production processes
Identification of process improvements

Table 2: Challenges reported

Recurring themes
Resistance from teams to adopting processes
Management monitoring
Institutionalization of the proposed improvements

the literature review, the client's needs, and the analysis of the systems' workflows to identify potential measures and derive quality indicators. We created a spreadsheet to organize these measures and indicators.

In the initial draft, we identified 133 measures. For each measure, four fields were described: identification (ID), description, potential related indicator, and a review field to record the person responsible for measure validation. We also included the data source for each measure, specifying whether it was identified from previous projects, literature, or client interviews.

In the **Categorization of Indicators** activity, we analyzed the first draft of measures and indicators and decided to organize them into five modules: *Management*, *Requirements*, *Development*, *Testing*, and *Security*. We started with the testing and requirements modules, as we identified 73 related measures. For each module, we revised the measures and indicators, removing all duplicates. During the refinement process, we also derived additional indicators. To facilitate the process, a new tab was created in the spreadsheet for each module. In this step, we added the ID for the indicator and a description. For each module, the indicators were structured into categories. For example, in the *Testing* module, six categories are defined: *Test Effort and Productivity*; *Fault Management*; *Test Coverage*; *Test Effectiveness and Efficiency*; *Test Automation* and *Testing Process*. We did not define a specific module for *Process*. Instead, we incorporated measures or indicators related to processes into other modules. For example, measures related to the testing process are categorized under the Testing module, within a category called *Testing Process*.

In the **Formulation of Indicator Metrics** activity, we analyzed each indicator to define how it would be calculated, *i.e.,* based on the underlying metrics. Mathematical expressions were defined by combining measures to interpret each indicator accurately. We conducted the *Validation* and *Implementation* processes incrementally for each module.

Table 3 shows examples of indicators, their related measures, and formulas. The definition of quality indicators is general. Thus, quality indicators can be measured at the Sprint, Release, or Project level, depending on the measurement goal. For example, IM15 (Average Time to Homologation) can be calculated based on MM46 and MM56, where MM46 can be collected at the Sprint or Release level. Similarly, IT27 (Rate of Successfully Executed Test Cases) can consider manual or automated execution in its calculation, and IR1 (Rate of Ambiguous Requirements) can consider requirements specified as user stories or in other notations.

Furthermore, quality indicators structured into a module can use measures from other modules in their calculation. For example, IT10 (Percentage of functional and non-functional requirements with automated tests in a Sprint or Release) is calculated as IT10 = (MT20/MR3)*100, where MT20 is the number of requirements with automated test case and MR3 (from Requirements module) is the quantity of requirements or user stories specified in a Sprint or Release.

4.4 Validation of Quality Indicators

Figure 4 presents an overview of the four activities of the *Validation of Indicators* subprocess.

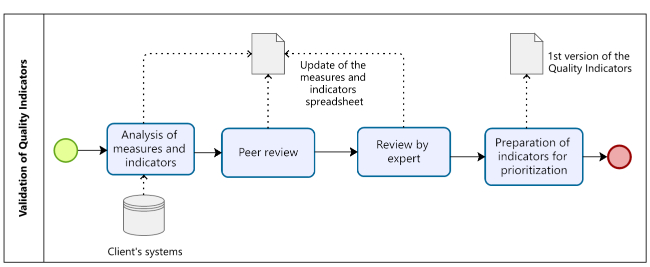

Figure 4: Subprocess for validating the quality indicators

Based on the initial set of measures and indicators, an internal validation was performed by the **Quality and Processes** team. First, the **Analysis of measures and indicators** was conducted to evaluate their relevance related to management decisions and the reliability of their calculations. Additionally, the descriptions of measures and indicators were improved to ensure clarity, and any inconsistencies were resolved. The team also performed the **Peer review**. Once a team member completed their review, another team member performed a second review, ensuring a proper peer review process. Next, the set of measures and indicators revised by the **Quality and Process** team was analysed by a domain expert during the **Review by expert** activity. For example, the measures and indicators of the Testing Module were validated by a test expert.

Once the expert review was completed, the suggested improvements were incorporated, and a first version of the measures and indicators was delivered for prioritization by the client's internal teams in the **Preparation of Indicators for Prioritization** activity. Thus, the **Quality and Processes** team organized a new spreadsheet for each module containing three tabs:

- The first tab, titled Instructions, provides an explanation of the spreadsheet and the module containing the target indicators to be evaluated.
- The second tab, Measures, presented the list of measures, each identified by an ID and a description.
- The third tab, Indicators, listed the indicators, organized by *category*, *ID*, *description*, and the corresponding metric used for calculation.

Additionally, we included a specific column to prioritize each indicator using three levels:

- *High*, indicating that the indicator is essential to the platform's objectives and must be included;
- *Medium*, indicating that the indicator is important but not critical; and
- *Low*, indicating that the indicator does not add value and should not be included in the platform.

An optional column labeled "Observation" was included for client's questions or suggestions.

4.5 Prioritization of indicators

The measures and indicators were submitted for analysis and prioritization by the client, represented by the contracting public institution's stakeholders. Using criteria that included each indicator's business value and implementation complexity, internal sessions were held within the internal organizational units of the company for collaborative prioritization, culminating in the selection of the final subset of indicators.

The result was a set of indicators suitable for integration into software management and development processes, ensuring that the organization's strategic decisions were supported by robust and relevant data. The first module prioritized was the Testing module, followed by the Management, Requirements and Development modules. The Security module has not been prioritized yet.

5 Findings and future directions

This project resulted in a structured catalog of quality indicators, specifically tailored to the software management and development practices of the partner public institution. A total of 159 quality

Table 3: Example of Quality Indicators per module

ID/Measure	Category	Indicator	Metric (formula)
Management			
MM46: Total deliveries submitted for approval **MM56**: Time taken for a delivery to enter homologation and reach production	Performance/ Productivity	Average Time to Homologation	IM15 = (SUM(MM56))/MM46
Requirements			
MR2: Number of requirements or user stories specified per project **MR21**: Number of ambiguous requirements or user stories **MR3**: Quantity of requirements or user stories specified in a Sprint or Release	Quality of Requirements	Rate of ambiguous requirements	IR1 = (MR21 / MR2) * 100
Testing			
MT1: Quantity of test cases executed **MT27**: Number of test cases executed successfully **MT20**: Number of requirements with automated test cases	Test Effectiveness and Efficiency	Rate of successfully executed test cases	IT27 = (MT27/MT1)*100
	Test Coverage	% of RF and NFR with automated tests	IT10 = (MT20/MR3)*100
Development			
MD5: Total time spent in feature implementation **MD6**: Total number of features delivered	Productivity/ Efficiency	Average Feature Delivery Time	ID10 = (MD5/MD6)

indicators, derived from 209 measures, were formally defined and documented. Key findings from the project include:

- **Indicator modeling:** As illustrated in Table 3, each indicator was formally defined with a unique identifier, an operational description, and a calculation formula, in accordance with established indicator modeling guidelines.
- **Indicator categorization:** The indicators were categorized into Project Management, Requirements, Development, and Testing modules to facilitate analysis and align with the client's main areas of interest.
- **Prioritization outcome:** The partner government institution actively participated in the process by prioritizing indicators specifically aligned with its quality monitoring objectives.
- **Tool-driven feasibility mapping:** The analysis of client tools (e.g., Assyst and ClickUp) facilitated the identification of concrete data collection points and viable metrics.

Tables 4, 5, 6, and 7 provide an overview of the indicator categories and the number of indicators defined for each module, demonstrating well-distributed coverage across management, requirements, development, and testing aspects.

Table 4: 40 Indicators per Management Module

Category	Total
Performance / Productivity	14
Planning / Monitoring	17
Cost and Effort	4
Management Process	3
Scope	2

Table 4 lists 40 management indicators organized into five categories, ensuring comprehensive coverage of project management performance. It reflects a strong focus on operational monitoring and project control. Table 5 presents 43 indicators focused on requirements quality and traceability across the development lifecycle, emphasizing the importance of managing changes effectively and maintaining consistent documentation.

Table 5: 43 Indicators per Requirements Module

Category	Total
Quality	2
Documentation Accuracy	9
Stability	5
Rework	3
Traceability	3
Requirements Process	10
Planning/Monitoring	6
Satisfaction	4
Documentation Completeness	1

Table 6 comprises 37 indicators that measure product reliability, maintainability, productivity, and efficiency. It emphasizes process optimization, code quality, and defect management. Table 7 details 39 testing indicators grouped into six categories. The emphasis is on assessing the completeness, efficiency, and automation of testing activities to ensure product reliability and continuous quality improvement.

Future work involves completing the definition of indicators for the Security module, which is still in progress, as well as defining the implementation process for the indicators.

Table 6: 37 Indicators per Development Module

Category	Total
Reliability	
Quality in Fault Detection and Correction	5
Product Quality - Maturity	1
Delivery effectiveness	3
Manutenibility	
Quality in Fault Detection and Correction	1
Internal Code Quality	9
Quality of the Code Review Process	2
Reusability	1
Modifiability	2
Analyzability	1
Productivity / Efficiency	8
Development Process	4

Table 7: 39 Indicators per Testing Module

Category	Total
Test Coverage	14
Test Effort and Productiviy	4
Failure Management	8
Test Effectiveness and Efficiency	4
Testing Process	2
Test Automation	7

6 Lessons Learned

This section presents the main lessons learned throughout the execution of the project, based on the challenges faced and the improvement opportunities identified while carrying out the activities.

(1) **Collaboration of client teams.** Throughout the process of defining the quality indicators, there was consistent and constructive collaboration between the client and the contracted teams. The client's availability to clarify doubts through meetings, interviews, answering surveys, and sharing of internal meeting recordings was essential to understand the processes and systems used by the organization. Furthermore, the active participation of the client in planning meetings, Sprint reviews, and strategic sessions greatly contributed to aligning project goals and ensuring that the defined indicators effectively addressed their needs.
(2) **Strategic Training and institutional partnerships.** The project offered training opportunities in several areas, including Data Quality and Software Quality, Requirements, Testing and Processes. These sessions enhanced team's technical skills and ensured conceptual alignment throughout the project. They also promoted outreach activities at the university and events linked to the Government Research Programs, contributing to the broader goal of overcoming the gap between academia and public administration.
(3) **Unavailability of process documentation from the external consultancy.** The absence of detailed documentation and insufficient information regarding the processes adopted by the external consultancy hindered a full understanding of the preceding context and limited the integration of efforts in the development of indicators.
(4) **Limited effectiveness of the Data Canvas workshop.** The Data Canvas workshop did not produce deliverables that effectively contributed to the project's progression, highlighting the need for pre-meeting preparation to ensure productivity.
(5) **Challenges in accessing institutional tools.** Difficulties in obtaining timely access to the tools used by the public institution introduced delays in project activities and impacted the planned timeline for conducting in-depth analyses of actual usage flows, which limited the understanding of operational processes.
(6) **Absence of a pilot project.** The lack of planning and preparation of a pilot project compromised the incremental validation of the proposed approaches, limiting opportunities for early adjustments before large-scale implementation.
(7) **Misalignment in the identification of sectors within the project scope.** Although several sectors of the public institution were initially considered, it was later identified that only one was effectively within the project's scope. This scenario led to efforts being directed toward areas that were not directly involved, highlighting the importance of early clarification regarding the sectors to be engaged.
(8) **Limited stakeholder engagement in the prioritization of indicators.** The indicator prioritization phase experienced lower-than-expected stakeholder participation, which may have affected the representativeness of the decisions and their alignment with the organization's specific needs, highlighting the importance of fostering broader engagement to ensure shared ownership and relevance of outcomes.

7 Threats to Validity

The main threats to validity concern the sources used to define the indicators, such as documents produced by the external consultancy within the public institution, and the selection of non-representative client interviewees. To mitigate these threats, we considered the experience of professionals responsible for operationalizing activities and making decisions. We also relied on technical literature and the analysis of relevant ISO standards (25010, 25022, 25023, and 25024) to support the formulation of measures, metrics, and indicators. Another threat concerns the prioritization of indicators, which was conducted with a limited number of internal organizational units of the company. This limitation was mitigated by selecting the organizational units that are most closely aligned with the goals of the indicators.

8 Conclusion

This report presented the process for defining and validating quality indicators within a public institution for over eight months. The definition of the indicators involved continuous collaboration between the **Quality and Process**, the **Requirements** and **Data Science** teams, and the client. A total of 159 quality indicators were defined, based on 209 measures. To support systematic cataloging and facilitate the client's prioritization process, these indicators were organized into five modules: *Management*, *Requirements*, *Development*, *Testing*, and *Security*. For each module, we defined measures, metrics, and quality indicators. Additionally, the indicators were classified either according to quality characteristics (*e.g.,* Maintainability and Reliability) or based on project-specific categories (*e.g.,* Testing, Development, or Requirements Processes). Each indicator was documented with an identifier, description, and a corresponding formula.

As future work, we intend to complete the definition of Security-related indicators and implement the prioritized indicators for data collection, integrated with the developed data quality platform. This implementation will involve standardizing the client's internal processes to enable systematic and consistent data collection within the data analyst pipeline.

ACKNOWLEDGMENTS

We thank Ceará Foundation for Scientific and Technological Development (FUNCAP) for their financial support. We also thank CNPq for the productivity grants to Prof. Rossana M. C. Andrade (306362/2021-0) and Prof. Ismayle S. Santos (312173/2025-3).

REFERENCES

[1] M. Alsulami. 2021. A Systematic Literature Review on Software Metrics. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies* 12, 12 (2021), 12A12G, 1–13. doi:10.14456/ITJEMAST.2021.238
[2] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. 1994. The Goal Question Metric Approach. In *Encyclopedia of Software Engineering*. John Wiley & Sons, 528–532.
[3] Victor R. Basili, Gianluigi Caldiera, and H. D. Rombach. 1994. The Goal Question Metric Approach. *Encyclopedia of Software Engineering* 2 (1994), 528–532.
[4] Rainara Carvalho, Anna Marques, Rossana Andrade, Amanda Sousa, Raquel Dias, and Guilherme Campagnoli. 2022. An Experience of Using the GQM Approach in a Remote Environment to Define Requirements Metrics. In *Anais do XXI Simpósio Brasileiro de Qualidade de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 219–228. https://sol.sbc.org.br/index.php/sbqs/article/view/23308
[5] Dave Chaffey. 2015. *Digital Business and E-commerce Management: Strategy, Implementation and Practice* (6 ed.). Pearson Education, Hoboken, NJ.
[6] CMMI Institute. 2010. *CMMI for Development, Version 1.3: Improving processes for better products*. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. Available at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9661.
[7] Alex Costa, Leidiana Freitas, Dafne Cavalcante, Victória Oliveira, Valéria Lelli, Ismayle Santos, Pedro Oliveira, Tales Nogueira, and Rossana Andrade. 2024. Especificação de Requisitos em um projeto de Big Data no Setor Público. In *Anais do XXVII Congresso Ibero-Americano em Engenharia de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 417–420. doi:10.5753/cibse.2024.28528
[8] Alistair Croll and Benjamin Yoskovitz. 2013. *Lean Analytics: Use Data to Build a Better Startup Faster*. O'Reilly Media, Sebastopol, CA.
[9] Thomas H. Davenport. 2013. Analytics 3.0. *Harvard Business Review* (Dec. 2013).
[10] Wayne W. Eckerson. 2011. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business* (2 ed.). John Wiley & Sons, Hoboken, NJ.
[11] Norman E. Fenton and James M. Bieman. 2014. *Software Metrics: A Rigorous and Practical Approach* (3rd ed.). CRC Press, Boca Raton, FL.
[12] José Finocchio Júnior. 2013. *Project Model Canvas: Gerenciamento de projetos sem burocracia*. Elsevier, Rio de Janeiro.
[13] Xavier Franch. 2021. Data-Driven Requirements Engineering: A Guided Tour. In *Evaluation of Novel Approaches to Software Engineering*, Raian Ali, Hermann Kaindl, and Leszek A. Maciaszek (Eds.). Springer International Publishing, Cham, 83–105.
[14] R. B. Grady. 1992. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
[15] Heli Helenius. 2024. Metrics Contributing to the Implementation of Successful Projects. (2024). Unpublished manuscript.
[16] International Organization for Standardization. 2011. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
[17] International Organization for Standardization. 2015. ISO/IEC 25024:2015 — Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of data quality.
[18] International Organization for Standardization. 2016. ISO/IEC 25022:2016 — Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of quality in use.
[19] International Organization for Standardization. 2016. ISO/IEC 25023:2016 — Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality.
[20] International Organization for Standardization. 2017. ISO/IEC 15939:2017 — Systems and software engineering — Measurement process. https://www.iso.org/standard/71197.html. Accessed: 2025-05-10.
[21] ISACA. 2012. *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*. ISACA, Rolling Meadows, IL.
[22] Newton Krüger Tallens Junior, Emerson Gomes dos Santos, Camila Bertini Martins, and Rogerio Scabim Morano. 2025. Construção de indicadores de desempenho organizacional para o setor público: atributos e critérios de validação em proposta inovadora. *Revista do TCU* 155, 1 (2025), 270–299. doi:10.69518/rtcu.155.270-299 Acesso em: 5 out. 2025.
[23] Liza Kayser, Roland M. Mueller, and Tizian Kronsbein. 2019. Data Collection Map: A Canvas for Shared Data Awareness in Data-Driven Innovation Projects. In *Proceedings of the 2019 Pre-ICIS SIGDSA Symposium*. 18. https://aisel.aisnet.org/sigdsa2019/18
[24] Barry Kitchenham, Emilia Mendes, and Gerson H. Travassos. 2007. A Systematic Review of Cross-Versus Within-Company Cost Estimation Studies. *IEEE Transactions on Software Engineering* 33, 5 (2007), 316–329. doi:10.1109/TSE.2007.1001
[25] OECD. 2020. *The OECD Digital Government Policy Framework: Six dimensions of a Digital Government*. OECD Public Governance Policy Papers 2. OECD Publishing, Paris. doi:10.1787/f64fed2a-en
[26] Alexander Osterwalder and Yves Pigneur. 2010. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons, Hoboken, NJ.
[27] US Poornima and V Suma. 2012. Significance of Quality Metrics in Software Development Process. *arXiv preprint arXiv:1207.0592* (2012).
[28] Roger S. Pressman and Bruce R. Maxim. 2016. *Engenharia de Software: uma abordagem profissional* (8 ed.). McGraw-Hill Brasil.
[29] Foster Provost and Tom Fawcett. 2013. *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media, Sebastopol, CA.
[30] K. Pulford, A. Kuntzmann-Combelles, and S. Shirlaw. 1995. *A Quantitative Approach to Software Management: The AMI Handbook*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
[31] Junaid Rashid, Toqeer Mahmood, and Muhamad Wasif Nisar. 2019. A study on software metrics and its impact on software quality. *arXiv preprint arXiv:1905.12922* (2019).
[32] Laura Cozzi Ribeiro, Matheus Pereira Libório, Hasheem Mannan, Sandro Laudares, Petr Iakovlevich Ekel, Douglas Alexandre Gomes Vieira, and Cristiane Neri Nobre. 2024. Software for building and measuring the quality of composite indicators using ordered weighted averaging: So-called S-CI-OWA. *SoftwareX* 26 (2024), 101660. doi:10.1016/j.softx.2024.101660
[33] Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE-PR). [n. d.]. Sebrae Canvas: ferramenta para criação de modelos de negócios digitais. https://sebraecanvas.com/. Acesso em: 11 jul. 2025.
[34] Ian Sommerville. 2011. *Software Engineering* (9 ed.). Addison-Wesley.
[35] Touseef Tahir, Ghulam Rasool, and Cigdem Gencel. 2016. A systematic literature review on software measurement programs. *Information and Software Technology* 73 (2016), 101–121. doi:10.1016/j.infsof.2016.01.014
[36] Rini Van Solingen and Egon W. Berghout. 1999. *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill.
[37] Richard Y. Wang and Diane M. Strong. 1996. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems* 12, 4 (1996), 5–33. doi:10.1080/07421222.1996.11518099
[38] Yichuan Wang and Nick Hajli. 2017. Exploring the Path to Big Data Analytics Success in Healthcare. *Journal of Business Research* 70 (2017), 287–299. doi:10.1016/j.jbusres.2016.08.002
[39] Élcio Batista, Rossana Andrade, Ismayle Santos, Tales Nogueira, Pedro Oliveira, Valéria Lelli, and Victória Oliveira. 2024. Fortaleza City Hall Strategic Planning based on Data Analysis and Forecasting. In *Anais do XXVII Congresso Ibero-Americano em Engenharia de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 433–436. doi:10.5753/cibse.2024.28532