Journal of Computer and Communications, 2023, 11, 18-43
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

ISO25000-Related Metrics for Evaluating the Quality of Complex Information Systems

Antonia Stefani, Bill Vassiliadis

Department of Management Science and Technology, University of Patras, Patras, Greece
Email: astefani@upatras.gr

How to cite this paper: Stefani, A. and Vassiliadis, B. (2023) ISO25000-Related Metrics for Evaluating the Quality of Complex Information Systems. *Journal of Computer and Communications*, **11**, 18-43.
https://doi.org/10.4236/jcc.2023.119002

Received: August 9, 2023
Accepted: September 17, 2023
Published: September 20, 2023

Abstract

Evaluating complex information systems necessitates deep contextual knowledge of technology, user needs, and quality. The quality evaluation challenges increase with the system's complexity, especially when multiple services supported by varied technological modules, are offered. Existing standards for software quality, such as the ISO25000 series, provide a broad framework for evaluation. Broadness offers initial implementation ease albeit, it often lacks specificity to cater to individual system modules. This paper maps 48 data metrics and 175 software metrics on specific system modules while aligning them with ISO standard quality traits. Using the ISO25000 series as a foundation, especially ISO25010 and 25012, this research seeks to augment the applicability of these standards to multi-faceted systems, exemplified by five distinct software modules prevalent in modern information ecosystems.

Keywords

Software Quality, Information Systems, Quality Standards, Quality Metrics, ISO25000

1. Introduction

Evaluating complex information systems is a complicated process that requires a comprehensive understanding of both the technologies and the requirements of the system's users, as well as the concept of quality. The latter is perceived from multiple perspectives and by different users. It's generally accepted that evaluating quality, even for a monolithic system regarding the aspects that users see, is challenging [1]. The assessment of quality, based on the factors perceived by users, partly depends on the type of services provided, which, in turn, rely on the technology implementing these services. Based on this reasoning, a complex information system, which offers services organized in (software) modules and technologically supported by distinct software segments, is even harder to evaluate [2].

The literature provides standards for evaluating software quality, which are horizontal, meaning they don't consider the type of services the software provides [3]. This is useful in terms of the practicality of the initial implementation of the standard, but it has the drawback of being too general. The evaluation provides general information that requires specialized processing or domain knowledge to become practical. This disadvantage is partially mitigated using software metrics, *i.e.*, measures linked to quality characteristics. In this way, an official standard increases its practicality, as measurements, whether quantitative or qualitative, provide more information to designers and evaluators, enabling them to design new or improve existing software. However, the drawback of generality is not eliminated. When linked to the quality characteristics of the software quality standard, metrics inherit these features' generality [4] [5].

In this paper, we design new metrics (and metametrics) and group metrics that already exist in the literature and connect them with the specific modules of an information system, retaining the link with the quality characteristics and sub-characteristics of the ISO25000 series standard. Based on previous related work [6] [7], we aim to further increase the practical application of official quality standards, specifically the ISO25000 series standard. To extend this hypothesis, we use a complex information system with five (5) software modules (Workflow Management, Data Warehouse, User Management, E-commerce, and Business Intelligence). Many modern information systems, with applications in e-commerce, enterprise resource management, human resource management, or process management, meet these needs.

We use as a base the ISO25000 series standard, also known as SQuaRE, an international standard related to the quality of software and information systems [8]. This particular standard establishes a software quality assessment framework, aiding producers and users in better understanding quality specifications. The ISO/IEC25000 encompasses several subcategories, each focusing on different aspects of software quality. The methodology of the research utilizes the ISO 25010 and ISO25012 standards, which are integral parts of the ISO25000 series standard. ISO 25010, entitled "Quality Models for Systems and Software," offers a comprehensive guide to the quality of software products, while ISO25012, referred to as the "Data Quality Model," focuses on the quality of information data. We believe its use is essential since many information systems focus more on data than processes. We provide a 2-dimensional mapping of 175 software metrics to ISO quality sub-characteristics and IS modules for a more focused quality assessment process. A set of 48 data metrics and 17 metametrics are also included in the study for completeness.

The structure of this paper is as follows: in Section 2, we provide insights into complex Information Systems, highlighting the difficulties of quality evaluation using formal standards, especially the ISO25000 series. Section 3 offers a more in-depth presentation of the ISO25010 and ISO25012 standards. Section 4 details the quality metrics for complex Information Systems, categorizing them according to quality characteristics and modules. Section 4 also presents meta-metrics and a discussion on the practical application of these metrics during the validation and verification of software. Concluding the paper, we reflect upon the contributions of this research and address its limitations.

2. Standards for Information Systems Evaluation

2.1. Complex Information Systems and Quality

As a hypothesis, we refer to a complex (modular) information system as an Information System (IS), which provides various diverse services organized in modules. The quality assessment of such a system is a complicated process due to the diversity of its components, possibly a multi-faceted approach, considering the individual characteristics of each module [9].

Each module in a complex information system may have its unique functionalities, user requirements, and potential pitfalls. Consequently, assessing the quality of each individual module can be likened to evaluating multiple distinct systems. Additionally, the interdependencies and interactions between these modules add another layer of complexity to the assessment. When one module malfunctions or underperforms, it may produce ripple effects throughout the entire system, impacting the efficiency and effectiveness of other modules. The metrics and benchmarks used for quality assessment may vary from one module to another. For instance, the module responsible for data storage might be evaluated based on its speed, capacity, and reliability. In contrast, a module designed for user interaction might be assessed based on user-friendliness, responsiveness, and accessibility.

In this paper, we argue that the ISO25000 series standard can be used to assess complex information systems more efficiently if suitable metrics are mapped to individual components. For this research, we use, as a case study, a model of a modular, general-purpose Information System with the following components (sets of services):

- Workflow Management: This service aims to unify the protocol assignment process and facilitate the traceability of protocolled documents throughout their life cycle. It entails a set of functions controlling the inflow and outflow of documents during their circulation. It will include document management and digital signature of documents (possibly using third-party software, the license and installation of which is the responsibility of the contractor).
- User Management: This service aims to offer a secure, consistent, and unified mechanism for managing the users and their roles for the entirety of the system. This service implements a secure access policy to the content and services of the system while providing efficient ways to prevent access to unauthorized users and limit the outcomes of malicious actions.
- Data Warehouse: The objective of this service is the gathering, standardization, organization, and utilizing data and operational knowledge derived from

primary data sources. It's a set of functions mainly targeting internal users, organizing, and managing data originating from various participant registries and historical data. It supports feedback mechanisms and data extraction applications.

- Business Intelligence: The goal of the service is to provide high-quality, uniform, and cohesive data to facilitate complex queries, monitor results, and assist decisions at both a tactical and strategic level. It concerns services which merge data (data fusion) from internal and external repositories and store them in an appropriate format in the Data Warehouse of the IS. This merging combines information stored in various heterogeneous environments, their integration, and presentation in a single, consistent business model. It provides the capability for query submission, conducting research, and producing reports based on dynamic criteria.
- E-commerce Portal: This service aims to enable online purchasing to shoppers and partners via a centralized portal. It aims to assist them in making informed purchasing or partnership decisions, leveraging the latest and most efficient e-commerce technologies.

The architecture of the IS conceptually groups services into subsystems (modules). The purpose of the subsystems is to integrate processes that will be used by the applications that constitute the IS. Each subsystem should be considered an autonomous entity but should cover a range of functions that are characteristic of it. For example, the user management subsystem should not depend on the data warehouse subsystem but only be interconnected. The development, upgrading, and maintenance should not prevent the other's upgrading, and maintenance. The subsystems should be central to the IS since many applications will depend on them.

The design, analysis, development, testing, evaluation, and support of complex IS, as defined in this work, is relatively ideal for complying with the software quality standard ISO25000 series (SQuaRE—System and Software Quality Requirements and Evaluation) [8]. The standard ISO/IEC 25012:2008 defines the general data quality standard for data stored within a system. It is used to determine the data quality requirements, the data quality measures, and the design and conduct of evaluations of the data quality of IS. The standard categorizes quality characteristics into fifteen characteristics from two perspectives: inherent and system dependent. It is intended for use in conjunction with other parts of the 25,000 series, such as the ISO/IEC25010 standard [10]. The data quality of an IS complies with ISO25012, which belongs to the ISO25000 series of standards [11]. This standard provides a general framework for evaluating the quality of data from various perspectives, including application requirements, product quality, and data quality during data management. The ISO25012 standard is ideal for the IS as it finds broad application in organizations and businesses that handle large amounts of data and want to ensure the high quality of their data [12]. Using the standard can help improve data processing performance, evaluate, and select data-based software or services, and ensure compliance with security and data protection requirements.

2.2. Information Systems Lifecycle Management with ISO Standards

The application of the ISO25000 series of standards enables the management of the life cycle phases of an IS (ISO25020) and its data (ISO25012). The primary principles are:

- Effectiveness: This refers to the efficient coverage of all life cycle phases. The standard addresses every phase of an IS life cycle, meaning many processes may be covered in each phase. Although covering the entirety of these processes is an advantage, it also increases the complexity of the IS.
- Completeness: This covers all modules of the IS, irrespective of the type of user, data, architectural specifics, and dissemination means. The standard encompasses two main axes: the operational and the technological.
- Flexibility: The standard allows for various certification levels. Following the proven best practices of the ISO organization, the ISO25000 series is hierarchical and non-overlapping. However, the border between sub-characteristics is often only apparent to specialists.
- Practicality: It responds to real-world needs by defining levels based on economic and operational parameters and the IS's cost-performance relationship. The practicality of the standard is enhanced by the use of quality metrics to provide tangible performance measurement indicators and best practices that offer more detail or clarification of the standard's guidelines.
- User-Centric Focus: A critical prerequisite for the successful use of an IS is user participation, both in design processes (through needs analysis) and in improvement (via evaluation). Many standards' shift towards a user-centric focus is emphatically expressed in various instances, either by adding to existing standards (like the Quality in Use pillar to ISO9126) or by designing the standard entirely based on the user (as with ISO25000).

The philosophy of ISO standards is to use different evaluation approaches in the same standard, depending on the product's life cycle phase or which part of the product is being evaluated. Each approach corresponds to specific, distinct features and sub-features.

In the new standards of the 25,000 series, Internal and External Quality and Quality in Use are linked to the phases of the product's life cycle. Based on this definition, quality (hence the components of a standard) can be related to the software lifecycle through the life cycle model. Quality is preferred (according to ISO25010) to three main phases of the product's life cycle:

- During the phase where the product is under construction: the evaluation refers to Internal Quality.
- During the phase where the product is in the evaluation phase: the evaluation refers to External Quality.

- During the phase where the product is in the usage phase: the evaluation refers to Quality in Use.

The requirements for Quality in Use are of great importance and it determines the required quality level from the user's perspective. These requirements are used to validate the product by users. They are determined by specific quality metrics in use. External requirements contribute to the recognition and definition of internal quality requirements and are, in turn, used to predict the requirements for quality in use. This creates a cycle where the requirements of one category contribute to determining the requirements of the next. On the other hand, Internal Quality requirements on the quality from the "internal" side of the product. They are used to determine the properties of intermediate products of the production process (software requirements, source code, etc.). They are used to define the properties and non-executable deliverables such as documentation and user manuals. They serve as validation targets at various stages of product development for determining development strategies, validation criteria, and evaluation. These requirements are quantitatively defined in the form of metrics—measures.

3. The ISO25010 Standard

3.1. The Series Standards

The role of standards is to provide guidelines for ensuring the quality of data or software. A standard is an official agreement that details technical specifications that can be used as rules for evaluating a subject [7].

The structure of ISO organization standards is usually hierarchical. At the top of the hierarchical structure are the quality characteristics. They constitute categories of quality components that do not overlap. Each characteristic contains (or is broken down into) a set of non-overlapping quality sub-features. The non-overlapping nature of the characteristics implies that the relationship between characteristics and sub-features is one to many. These two levels of a standard's structure describe, in general terms, the quality components to which absolute values cannot be attributed during the evaluation of the subject, only descriptive values. This is necessary to ensure the generality of the standards, meaning their independence from specific techniques or implementation technologies of the evaluated object [10].

The structure's third level consists of metrics with also a one-to-many relationship with the sub-features. Metrics can take absolute values and are measures of quality. In many cases, their practical value is significant as they can provide precise information/guidance for the design/construction of quality objects. However, a numerical value cannot accurately reflect reality, as this holds for most absolute measures. Therefore, in quality evaluation, metrics should be used with caution.

Implementation guidelines or usage examples usually accompany standards. These do not constitute part of their structure. It is common for new standards that exclusively contain specifications or application instructions of other standards (that include quality models), management of the standard application processes, or general reference frameworks. If appropriately adapted, these can form the basis for quality evaluation systems or quality specifications.

The Software Quality Measurement Model, as defined in ISO25010, outlines the inherent properties of the software, which can be distinguished quantitatively or qualitatively as characteristics. Quality characteristics are the inherent properties of the software that contribute to its quality. These quality characteristics are categorized into one or more sub-characteristics. Quality characteristics are measured using a measurement method. The result of applying a measurement method is called a quality measurement element. Quality characteristics and sub-features can be quantified by applying measurement functions to these elements. A function is essentially an algorithm used to combine elements. The result of applying a measurement function is called a quality measure. In this way, quality measurement elements become quantified reflections of quality characteristics and sub-characteristics. More than one measure can be used to measure a feature or sub-feature.

3.2. About the ISO25000 Series Standard

The ISO25000 series, also known as Software Product Quality Requirements and Evaluation (SQuaRE), is the newest version of standards for software system quality. It was designed to replace the standards of the 9000 and 10,000 series with the goal of standardization and the elimination of overlaps. The standard's objective is to replace ISO9126-1 in terms of providing a quality model for evaluating software systems and services. ISO25010 is based on the quality model of ISO9126, has a similar hierarchical structure, and most of its characteristics and sub-characteristics are the same. The new standard does not have defined metrics and relies on the metrics of the ISO9126 standard.

The research methodology uses ISO25010 (which focuses on the quality of software and systems) and ISO25012 (which focuses on data quality). Both are crucial for evaluating and improving software and data quality [12]. ISO25010, entitled "Quality Models for Systems and Software," offers a comprehensive guide to the quality of software products, defining the main aspects of quality that need to be considered. It covers various elements such as performance, reliability, usability, and more. ISO25012, referred to as the "Data Quality Model," focuses on the quality of information data. It provides a thorough framework for data quality assessment, considering various perspectives, from application requirements to data management. Each of these characteristics is further broken down into sub-characteristics. This standard defines three categories of data quality: intrinsic data quality, system-dependent data quality, and user-dependent data quality. It evaluates various properties of data, such as accuracy, completeness, consistency, reliability, timeliness, accessibility, etc.

3.3. Quality Model Division with ISO25010

The goal of the ISO25000 series standard is to replace ISO9126-1 in terms of providing a quality model for the evaluation of software systems and services. The new standard does not have predefined metrics and relies on the metrics of the ISO9126 standard. The innovation of the present work is based on this observation. The ISO25010 standard addresses "System and Software Quality Models". It defines the quality characteristics and sub-characteristics that must be taken into account in the evaluation of a software product. The differences between the two standards are as follows:

- ISO25010 has eight features against 6 of ISO9126 and 39 sub-features.
- Functional completeness has been added as a sub-characteristic, and Interoperability and Safety have been moved as new quality features.
- The Accuracy has been renamed to Functional Correctness and the Suitability to Functional Suitability.
- Efficiency has been renamed to Efficiency Capability. The capacity is added as its sub-characteristic. Compatibility is a new feature that now includes sub-characteristics Coexistence (which moved from Portability) and Interoperability (which moved from Functionality).
- Usability has as new sub-features the User Error Protection and Accessibility (used by individuals with a wide range of characteristics).
- Understandability is renamed to Recognizability, and Attractiveness is renamed to User Interface Aesthetics.
- Reliability has a new sub-feature, that of Availability (when required to be used).
- Security is a new feature with sub-characteristics of Privacy (data accessible only by authorized users), Integrity (protection from unauthorized modification), Non-Repudiation of Responsibility and Authenticity Ability.
- Maintainability has new sub-features of its Extensibility.
- Reusability Replaceability and stability are components of Modifiability.

3.4. Data Quality with ISO25012

The ISO/IEC25012: 2008 standard defines the general data quality standard for data stored in a structured format within a computer system, such as an IS [11]. It can be used for defining data quality requirements, for data quality measurements, or for designing and conducting data quality evaluations. It could be used, for instance, to define data quality requirements during production, acquisition, and completion processes, to identify quality assurance criteria that are useful for the reuse, validation, and improvement of data, and for the reorganization, evaluation, and improvement of data, and to assess the compliance of the data with legislation or/and requirements. The standard categorizes quality characteristics into fifteen characteristics from two perspectives: inherent and system dependent. The ISO25012 data quality model defines:

- Internal Data Quality, which refers to:
  - Data values
  - Data types and sizes

- Data definitions (including metadata)
- Data rules
- Data links

- External Data Quality refers to the ability of the data to meet specified needs under specified conditions within a software system.

4. Quality Metrics for Complex Information Systems

4.1. Measures and Metrics

Metrics serve as a principal constituent feature of quality standards, positioned at the lowest level of the hierarchy. Nonetheless, their practical value is significant. Metrics are deployed for the appraisal and quantification of properties of the item or information under assessment. They present an empirical, objective assignment of a numerical or symbolic value to an entity, or a component of the assessed object aimed at evaluating a particular attribute of it. The utilization of metrics targets addressing the fundamental challenge of defining measurable quantities. While the notions of quality characteristics and sub-characteristics are marked by flexibility and breadth in interpretation, metrics are determined through measurements. Measurement is the process through which numbers or symbols are aligned with properties of the components comprising the evaluated system or product, describing them based on defined rules. The interpretation of measurements attributed to metrics remains a pivotal area of study, especially their interpretation concerning the quality of the system. The interpretation of a metric lies in determining its degree of correlation with one or more external features of the system.

The metrics of the ISO25000 series can be classified according to the nature of the object they evaluate or the type of feature they address. They can be discerned into process metrics, which pertain to the development process; resource metrics, addressing the available resources for the development of the object; and product metrics, focusing on the characteristics the system possesses for end-user delivery. A fundamental categorization of metrics is executed based on the nature of the features they measure, distinguishing between internal and external metrics.

There are two general categories of metrics. Internal metrics can be applied during the design and development phase. During the development process, interim derivatives can be assessed using internal metrics. The primary goal of internal metrics is to ensure the required external quality of the system and its quality in use. Internal metrics measure the system's internal features by analyzing the properties of the intermediates or deliverables before they are used under real conditions. Measurements of internal metrics refer to numerical data related to the frequency of appearance of the elements that constitute the object and, for example, refer to the source code, the flow diagram, and the system's complexity. Characteristic examples of internal metrics for software are the lines of code (LOC) that characterize the size of the software's source code and the cyclomatic complexity, a measure of the software's complexity based on the flow chart.

Cyclomatic complexity aims to highlight parts of the software that will be difficult to understand, test, and maintain. External metrics on the other hand, are based on the definition of quality that emphasizes user satisfaction and directly measure the desired external features and are grouped by quality sub-characteristic, suggesting a way to apply them. Their key feature is that they require user participation and developer involvement. They capture the system's external quality in combination with the internal knowledge of the features the system provides. The importance of external metrics is to depict the object's external quality in relation to the functions and services they offer to the end-user under real usage conditions.

4.2. Data Metrics per Quality Characteristic

Data Quality metrics can be produced per quality characteristics and be applied horizontally, that is, for all the modules of a complex Information system. Table 1 provides a non-exhaustive list of the data quality metrics and the corresponding sub-characteristic of ISO25012.

4.3. Metrics per System Component

4.3.1. Workflow Management

The ISO/IEC25010 standard presents a model for software product quality and encompasses various quality characteristics. When it comes to assessing a specific component through metrics such as Workflow Management in a complex IS (Table 2), it is important to consider how this module interacts with others and its primary functions.

4.3.2. User Management

The User Management component of an information system is crucial. This module typically encompasses functionalities related to user creation, modification, deletion, rights and permissions assignment, profile management, and authentication. Table 3 presents a detailed set of software metrics tailored for the User Management component based on ISO/IEC25010.

4.3.3. Data Warehouse

Assessing the Data Warehouse component is vital, given its role in consolidating, storing, and making available large volumes of data for querying and reporting purposes. Table 4 presents a set of software metrics tailored for this component based on ISO/IEC25010.

4.3.4. Business Intelligence

The Business Intelligence (BI) component is also fundamental in a complex information system, offering data visualization, reporting, analytics, and often machine learning capabilities. Table 5 presents software metrics tailored for this component based on ISO/IEC25010.

4.3.5. E-Commerce

Diversifying metrics allows for multi-dimensional analysis, ensuring all facets of

Table 1. Data metrics based on ISO25012.

ISO25012 Sub-characteristic	Data Metrics
Accuracy	• Data Error Rate: Percentage of data records that contain errors relative to the total number of data records. • Percentage of correct values: Percentage of data records that contain correct values relative to the total number of data records. • Total Data Loss Rate: Percentage of lost or unrecorded data records relative to the total number of data records. • Average Deviation: The average amount of deviation between the data values and the correct values.
Comprehension	• Simplicity of data structure: This can be measured as the percentage of data elements that are straightforwardly understood without additional explanation or interpretation. • Data consistency: Data consistency refers to the extent to which related pieces of data are coherent and consistent. • Quality of documentation: This can be assessed by the completeness, accuracy and comprehensibility of the documentation accompanying the data. • Clarity of tags and metadata: This can be assessed based on how easily one can understand the meaning of the data based on the tags and metadata that accompany it. • Clarity of units of measurement: This refers to the ease with which users can understand the units of measurement used for the data.
Consistency	• Inconsistency rate: This can be measured as the percentage of data that is not consistent or does not follow the rules or specifications that have been set. • Deviation from the norm: This can be calculated as the average deviation of the data from a certain norm or standard. • Number of Duplicates: The number of repeated entries in the data set. • Value Variance: The variance of the values of the data provided, which may indicate inconsistency if the values deviate greatly from the expected value.
Acceptability	• User satisfaction rate: This can be measured using questionnaires, ratings or reviews conducted with users of the data. • Task execution success rate: Percentage of tasks that can be completed successfully using the data. • Data rejection rate: Percentage of data that is rejected or not used by users due to insufficiency, inaccuracy, or other quality issues. • Rate of data quality reports: Percentage of reports made about data quality, such as errors, inconsistencies, or other problems.
Reliability	• Validity Rate: It can be calculated as the percentage of data that meets the validity criteria set. For example, the validity of email addresses or phone numbers. • Percentage of Missing Values: It can be calculated as the percentage of missing or unavailable data. • Deviation from the predicted variance: This can be calculated as the deviation of the observed data from the predicted variance based on the estimates or predictions. • Retry Failures: Number of times data is not reliably retried under different conditions or to different users.
Efficiency	• Return on Investment (ROI): This can be calculated based on the value or return achieved from using the data compared to the cost of accessing, processing, maintaining, and analyzing the data. • Performance Rate: It can be measured as the percentage of successful tasks performed using the data relative to the total number of tasks. • Processing Time: The time required to process or analyze the data. • Response Time: The time it takes to get results or information from the data.

Continued

Completeness	• Percentage of Missing Values: It can be calculated as the percentage of missing or unreported data. • Number of Missing Fields: This can be calculated as the number of data fields that have no values or content. • Number of Complete Entries: This is the number of entries in the database that are complete, *i.e.*, no data is missing. • Percent Complete: This is calculated as the percentage of data that meets the specifications or standards set for completeness.
Availability	• Downtime: This can be calculated as the total time that data was unavailable due to outages or disruptions. • Availability Rate: This can be calculated as the percentage of time that the data was available for use relative to the total time. • Recovery Time: The time required to recover data after an outage. • Rate of Successful Requests: It can be measured as the percentage of successful requests to access the data relative to the total number of requests.
Clarity	• Accuracy Rate: It can be measured as the percentage of data that is accurate relative to the total amount of data. • Number of Errors: This is the number of errors or inaccuracies found in the data. • Error Rate: This is calculated as the percentage of data that contains errors or inaccuracies relative to the total amount of data. • Comprehensibility Measurement: Can be done by assessing how easily a user can understand the data.
Distinctive ability	• Percentage of Unique Values: This can be calculated as the percentage of data that is unique relative to the total amount of data. • Value Range: This is the range of different values that the data can take. • Number of Categories: This is the number of different categories or classes into which the data can be classified.
Ease of learning	• Learning Time: This is the time required for a user to understand and use the data for the first time. • Learning Error Rate: This is calculated as the percentage of errors a user makes while learning and using the data. • User Evaluation: This can result from users' direct evaluation of how easy or difficult it was to understand and use the data. • Task Completion Rate: It can be measured as the percentage of tasks successfully completed by users after learning the data.
Flexibility	• Reshaping Rate: Measures the percentage of data that had to be changed or reshaped to adapt to new requirements or conditions. • Remodelling Time: This is the time required to remodel or modify the data to suit new requirements or conditions. • Number of Reformats: This is the number of times the data had to be reformatted or modified to meet new requirements or conditions. • Reusability: This metric refers to the ability of data to be reused under different conditions or in different contexts.

the E-commerce component's performance and user experience are addressed. Leveraging these metrics can offer actionable insights for system optimization and enhancement. **Table 6** provides a set of metrics (either simple or complex

Table 2. Workflow management module metrics based on ISO25010.

ISO25010 Sub-characteristic	Metrics
Functionality	• Protocol compliance: Whether the protocol aligns with intended standards. • Functional completeness: Coverage of the protocol functions compared to requirements.
Reliability	• Protocol uptime: The time the protocol operates without failure. • Message delivery success rate: Percentage of messages correctly sent and acknowledged.
Usability	• Protocol documentation quality: Ease of understanding and clarity of protocol documentation. • Error messages clarity: Descriptiveness and helpfulness of error messages.
Efficiency	• Latency: Time taken for a message to traverse the protocol. • Throughput: Number of messages handled per unit of time.
Security	• Encryption quality: Strength and type of encryption used. • Authentication attempts: Number of failed vs. successful authentications.
Maintainability	• Protocol modularity: Ease of isolating and modifying individual parts of the protocol. • Code complexity: Measured using metrics like cyclomatic complexity for any implemented parts.
Portability	• Interoperability: Ability of the protocol to interact with other systems. • System dependencies: Number and type of external systems or tools the protocol relies on.

and mostly technical in nature) for the E-commerce module.

4.4. Metametric Evaluation for Business Intelligence Components

A metametric is a metric that combines two or more existing metrics to measure complex or critical features. They provide a higher-level view by combining existing metrics. They can offer insights that might not be immediately apparent from individual metrics [13]. As a case study for the IS at hand, a set of Meta-metrics for the BI component based on the previously mentioned metrics is depicted in Table 7. It must be noted that these meta-metrics offer a holistic view of the BI component's capabilities and strengths. By blending individual metrics, it is possible to provide to IS stakeholdersa more comprehensive understanding of specific complex aspects of the system. They are especially useful when making strategic decisions or comparing multiple BI systems. However, the weights and formulas can be adjusted depending on the particular emphasis and priorities of the organization.

The utilization of meta-metrics to assess the Business Intelligence (BI) component offers an insightful perspective into the multi-faceted nature of such a module [9]. The right combination of multiple individual metrics, meta-metrics provide a comprehensive lens through which the system's performance and capabilities can be observed. More specifically, assessing a BI component using meta-metrics facilitates a holistic overview of the system. Rather than navigating the intricacies of each individual metric, decision-makers can swiftly discern the

Table 3. User Management metrics based on ISO25010.

ISO25010 Sub-characteristic	Metrics
Functionality	• Functional completeness: Coverage of user management functions compared to requirements (e.g., CRUD operations, password reset). • Functional correctness: Percentage of test cases passed related to user management functions. • Feature coverage: Percentage of user management features implemented compared to total requirements. Role granularity: Number of distinct user roles or profiles supported.
Reliability	• Error rate: Number of failed user operations (like login failures) over a specific period. • Availability: System uptime and availability for user management tasks. • Session reliability: Percentage of user sessions without interruptions or unexpected logouts. • Password reset success rate: Percentage of successful password resets compared to total reset requests.
Usability	• User onboarding time: Time taken for a new user to set up and understand the system. • -Intuitiveness: Survey-based metric on how easy users find the system to manage their profile and settings. • Dashboard clarity: User feedback or rating on the clarity and usefulness of user management dashboards. • Navigation efficiency: Average number of clicks or actions to perform common user management tasks.
Efficiency	• Response time: Time taken to process user management tasks like user creation or role assignment. • System resource utilization: Resources (like CPU, memory) used during intensive user operations. • Batch processing time: Time taken to process batch user operations (e.g., bulk user creation or deletion). • Session initialization time: Average time taken to initialize a user session upon login.
Security	• Authentication security: Strength and type of encryption and hashing used. • Role misassignment rate: Incidents where users receive incorrect roles or permissions. • Multi-factor authentication (MFA) usage: Percentage of users enrolled in MFA. • Password policy strength: An index or score based on the rigour of the password policy (e.g., length, complexity requirements).
Maintainability	• Code modularity: Ease of isolating and modifying parts of the user management component. • Code complexity: Metrics such as cyclomatic complexity or lines of code for the user management module. • API versioning frequency: How often user management-related APIs undergo version changes. • Database schema stability: Frequency of changes to the user-related database schema.
Portability	• Integration ease: How straightforward it is to integrate the user management module with other systems. • Database independence: The ability of the user management system to operate across different database platforms. • Cross-platform support: Number of platforms (e.g., mobile, web, desktop) supported by user management functions. • Integration flexibility: Ease and number of integrations with third-party systems (like SSO providers).
Interoperability	• SSO (Single Sign-On) support: Number of SSO protocols supported (e.g., SAML, OIDC). • Directory services integration: Ability to integrate with directory services like LDAP or Active Directory.
Auditability	• Change logs availability: Presence and completeness of logs capturing user profile changes. • Access logs retention: Duration for which user access logs are retained and easily retrievable.

Table 4. Data warehouse module metrics based on ISO25010.

ISO25010 Sub-characteristic	Metrics
Functionality	• Data coverage: Percentage of enterprise data domains covered by the data warehouse. • Data freshness: Frequency or timeliness of data updates. • Data lineage tracking: Ability to trace data from its source to its destination in the warehouse. Data redundancy: Percentage of redundant data or repeated information.
Reliability	• Data availability: Uptime or accessibility of the data warehouse. • Data accuracy: Percentage of records without discrepancies when validated against source systems. • Backup frequency: How often backups of the data are made. • Backup recovery success rate: Percentage of successful data recoveries from backups.
Usability	• Query simplicity: Average complexity or length of typical queries (can be used to gauge the structure and organization of the data). • Documentation quality: Completeness and clarity of data dictionaries, ETL (Extract, Transform, Load) process descriptions, and entity-relationship diagrams. • Metadata quality: Completeness and accuracy of metadata that describes the data. • User-friendly interfaces: Number of training hours required for new users to proficiently query the warehouse.
Efficiency	• Query response time: Average time taken to execute standard complex queries. • ETL process time: Time taken for data to be extracted, transformed, and loaded into the warehouse. • Storage efficiency: Ratio of data storage used to the total storage capacity. • Indexing efficiency: Time taken to index new data and speed improvements from using those indexes.
Security	• Data encryption: Strength and type of encryption used for data at rest and in transit. • Access violations: Number of unauthorized access attempts or breaches. • Audit trail capabilities: Availability and quality of logs for user access and data modifications. • Data masking: Percentage of sensitive data fields that are masked or anonymized.
Maintainability	• ETL modularity: Ease of modifying or adding new ETL processes without affecting existing ones. • Schema change frequency: Rate at which the data warehouse schema or structure changes, indicative of stability. • Change propagation time: Time taken to reflect changes from source systems in the warehouse. • Deprecation rate: Rate at which old data structures or fields are deprecated or become obsolete.
Portability	• Data exportability: Ease with which data can be exported into different formats or to different platforms. • Integration capabilities: Number and flexibility of interfaces or APIs available for connecting external systems to the data warehouse. • Cross-platform compatibility: Ability of the data warehouse to be migrated or to operate across different hardware or software platforms. • Data format diversity: Number of data formats (CSV, Parquet, Avro, etc.) that the warehouse can natively handle.
Performance	• Load scalability: How well the system performs as the volume of data increases. • Concurrency: Number of simultaneous queries or operations the system can handle without significant degradation in performance.
Operability	• Monitoring tools integration: How well the data warehouse integrates with monitoring and alerting tools. • Automated health checks: Frequency and coverage of automated system health checks.

Table 5. Business intelligence module metrics based on ISO25010.

ISO25010 Sub-characteristic	Metrics
Functionality	• Report Coverage: Percentage of business requirements covered by available reports. • Analytics algorithm accuracy: Accuracy of predictions or recommendations generated by BI tools. • Data source compatibility: Number of different data sources (like SQL, NoSQL, APIs) BI can natively connect to. • Advanced analytics support: Availability of features like predictive analytics, anomaly detection, and trend analysis. • Ad-hoc query support: Ability for users to create and run their own queries without relying on predefined templates or reports. OLAP capabilities: Support for Online Analytical Processing operations like slice-and-dice, drill down/up, etc.
Reliability	• Data refresh rate: Frequency of data updates in BI dashboards. • Report generation success rate: Percentage of reports generated without errors. • Historical data accuracy: Consistency of historical data representations over time. • Scheduled report reliability: Percentage of scheduled reports that run and deliver as expected. • Data transformation accuracy: The integrity of data when transformed from its raw form to a more structured or aggregated form for BI purposes. • Alert accuracy: Accuracy of automated alerts based on certain business conditions or thresholds.
Usability	• Dashboard intuitiveness: User feedback or rating on the clarity and usefulness of BI dashboards. • Custom report creation ease: Time and steps required for users to create custom reports. • Visualization variety: Number of different visualization types (bar charts, pie charts, heatmaps, etc.) supported. • Interactive capabilities: Ability of end-users to drill down, slice, or interact with reports dynamically. • Template availability: Number of predefined report and dashboard templates available for different business scenarios. • Guided analytics: Availability of guided or suggested analytical paths for users based on their objectives.
Efficiency	• Query response time: Average time taken to execute standard complex queries within the BI tool. • Load scalability: How well the BI system performs as data volume or user count increases. • Data caching efficiency: Reduction in report generation time due to caching mechanisms. • Compression efficiency: Storage savings from data compression without sacrificing query performance. • Data ingestion speed: Rate at which new data is ingested into the BI system. • Load balancing efficiency: Effective distribution of computational tasks across servers or nodes for optimized performance.
Security	• Data masking and anonymization: Percentage of sensitive data fields that are masked or anonymized in reports. • Access control granularity: Level of detail at which access rights can be specified (e.g., by report, by data field). • Audit trail quality: Completeness and clarity of logs capturing report access, data modifications, and user activities in the BI system. • Row-level security: Ability to restrict data access at a granular row level based on user roles or attributes. • Field-level security: Ability to restrict access to specific fields within a dataset, not just rows or entire datasets. • Data encryption standards: Level and methods of encryption for data at rest and in transit.

Continued

Maintainability	• Dashboard modifiability: Ease with which existing dashboards can be modified. • Data source integration flexibility: Ease of adding new data sources to the BI system. • Visualization library extensibility: Ease with which new visualization types can be added. • Metadata management simplicity: Effort required to maintain and update metadata that describes datasets, reports, or visualizations. • Automated error detection: The ability of the system to automatically detect and, if possible, correct errors in data or computations. • Version control: Mechanisms in place for versioning reports, dashboards, and data models.
Portability	• Export format diversity: Number of formats (e.g., CSV, PDF, Excel) in which reports can be exported. • Cross-platform compatibility: Number of platforms (e.g., mobile, web) on which BI tools are accessible and functional. • Cloud readiness: Ability of the BI component to operate and scale on cloud platforms. • Mobile responsiveness: Quality of BI dashboards and reports on mobile devices in terms of layout, interaction, and load time. • BI tool migration capabilities: Ease with which BI content (like reports, dashboards) can be migrated to another tool or platform. • Offline access: Capability to access certain BI features or content offline.
Performance	• Real-time processing: Latency between data ingestion and its availability in BI reports. • Concurrent user handling: Number of simultaneous users the system can support without significant performance degradation. • Background task speed: Speed at which background tasks (like data refreshes or scheduled report runs) are completed. • Aggregation speed: Time taken to aggregate or roll-up data at different levels. • Large dataset handling: Performance consistency when handling exceptionally large datasets. • Resource optimization: Efficiency in using computational resources like CPU, memory, and storage.
Interoperability	• Third-party tool integration: Number and types of third-party tools (e.g., CRM, ERP) that can be integrated with the BI component. • API availability: Availability and completeness of APIs for external system interactions. • Data connector extensibility: Flexibility in adding connectors to new, unsupported data sources. • Embedding capabilities: Ability to embed BI reports or visualizations in other applications or platforms. • Third-party visualization support: Ability to integrate or use visualization components from third parties. • Open standards adherence: Adherence to open standards for data connectivity, visualization, etc., promoting interoperability.
Compliance	• Data governance adherence: Compliance of the BI tool with organizational data governance policies. • Regulatory compliance support: Features that assist in complying with relevant data-related regulations (like GDPR or HIPAA).
Scalability	• Cluster scalability: How well the system scales out by adding more nodes or servers to a cluster. • Data growth adaptability: Performance consistency as the underlying data grows over time.
Customisability	• Plug-in or extension support: Ability for developers to add custom functionalities or integrations. • User-defined function support: Capability for users to define their own functions for specific computations.

Table 6. E-Commerce module metrics based on ISO25010.

ISO25010 Sub-characteristic	Metrics
Functional Suitability	• Percentage of complete transactions out of total attempted transactions. • Number of product listings that are correctly categorized. • Percentage of product returns due to listing inaccuracies. • Number of successful integrations with new payment gateways in a year. • Conversion Funnel Effectiveness: A measure derived from the ratio of users completing a purchase to those starting one, indicating the effectiveness of the purchasing process. • Feature Utilization Rate: A measure of how often specific functionalities (e.g., wishlist, cart add) are used compared to their availability.
Performance Efficiency	• Average response time during peak usage hours. • Throughput: Number of transactions processed per minute during sales events. • Page load time for high-resolution product images. • Percentage decrease in page load times post-optimization efforts. • Average Page Load Time Across User Geo-Locations: A metric that combines server response times, content rendering, and geographic disparities to ensure users across regions receive optimal performance.
Compatibility	• Number of integration-related errors per month. • Percentage of successful data synchronizations with integrated systems. • Number of successful API calls between the E-commerce platform and third-party services. • Number of customer complaints due to payment gateway integration issues. • Browser and Device Compatibility Score: Aggregate metric considering error rates, page rendering inconsistencies, and functionality failures across various devices and browsers.
Usability	• Average number of clicks to complete a purchase. • Percentage of users who abandon their cart before finalizing a purchase • Average rating from user feedback on the checkout experience. • Percentage of users who utilized product recommendations. • Interface Latency: Time taken for the user interface components to respond to user interactions, like button clicks or form submissions.
Reliability	• System uptime percentage over a given period. • Mean time between failures (MTBF). • Number of transactions rolled back due to system errors. • Duration of longest uninterrupted service uptime.
Security	• Number of security breaches or vulnerabilities detected in a given timeframe. • Average time taken to detect and mitigate a security breach. • Percentage of transactions conducted over a secured connection (e.g., SSL). • Number of unauthorized access attempts detected and thwarted.
Maintainability	• Average time taken to implement new features or updates. • Number of reported issues post-software updates or patches. • Number of deprecated features still in use by customers. • Frequency of code refactoring or optimization initiatives in a year.
Portability	• Time and resources needed to migrate to a new server environment. • Percentage of successful data migrations during platform changes. • Time is taken to adapt the E-commerce platform for a new region or market. • Percentage of modules successfully ported to a mobile app without major changes.

Table 7. Meta-metrics for the BI module based on ISO25010.

Meta-metric	Formula	Measurement Goal
Data Reliability Index	(Data refresh rate + Data transformation accuracy + Historical data accuracy)/3	Measures the overall reliability and integrity of data presented in the BI tool.
User Experience Score	(Dashboard intuitiveness + Custom report creation ease + Template availability)/3	Assesses the overall usability and user-friendliness of the BI system.
Security Robustness Rating	(Data masking and anonymization + Field-level security + Data encryption standards)/3	Gauges the overall security standards and practices of the BI tool.
System Efficiency Score	(Query response time + Data caching efficiency + Resource optimization)/3	Evaluates the performance efficiency of the BI system from a user's perspective.
Flexibility Index	(Ad-hoc query support + BI tool migration capabilities + Plug-in or extension support)/3	Measures the adaptability and extensibility of the BI system.
Interoperability Index	(Third-party tool integration + Data connector extensibility + Open standards adherence)/3	Assesses how well the BI tool can integrate with other systems and standards.
Comprehensive Security Score	(Audit trail quality + Alert accuracy + Row-level security)/3	Measures the depth and breadth of security features in the BI tool.
Optimization Index	(Load balancing efficiency + Large dataset handling + Cluster scalability)/3	Assesses the system's capacity to optimize and scale based on demand and data volume.
Data Integration Capability Score	(Data source compatibility + Data ingestion speed + Metadata management simplicity)/3	Measures how efficiently the BI tool can integrate and manage various data sources.
Analytical Depth Index	(Advanced analytics support + OLAP capabilities + User-defined function support)/3	Assesses the depth and sophistication of analytical functionalities in the BI tool.
User Empowerment Score	(Ad-hoc query support + Interactive capabilities + Guided analytics)/3	Gauges how well the BI system empowers users to derive insights on their own.
Performance Robustness Index	(Load scalability + Background task speed + Real-time processing)/3	Assesses the BI system's consistency in performance across varying demands.
Customization & Extensibility Score	(Dashboard modifiability + Plug-in or extension support + Visualization library extensibility)/3	Measures the system's flexibility in adapting to unique and changing requirements.
Collaboration Index	(Version control + Report sharing capabilities + Concurrent user handling)/3	Evaluates the collaborative features of the BI system among users.
System Resilience Rating	(Scheduled report reliability + Alert accuracy + Automated error detection)/3	Assesses the BI system's ability to operate flawlessly and recover from issues.
Mobility and Access Score	(Mobile responsiveness + Offline access + Cross-platform compatibility)/3	Measures how accessible and functional the BI system is across different devices and scenarios.
Compliance and Governance Index	(Data governance adherence + Regulatory compliance support + Audit trail quality)/3	Assesses the BI system's alignment with regulatory and organizational governance standards.

performance domains of the system. This approach also lends itself well to comparative analysis, enabling organizations to calibrate their BI system's performance against industry standards or rivals using universally recognized meta-metrics. Furthermore, when confronted with the need to make strategic decisions about areas of investment, meta-metrics provide some guidance on whether the emphasis should be on enhancing performance, user experience, or perhaps security.

However, the adoption of meta-metrics is not devoid of challenges. A notable concern is the potential for over-generalization. The very act of blending several metrics might inadvertently lead to the simplification of specific aspects, creating a blind spot for particular issues. Additionally, the question of weight allocation for each embedded metric within a meta-metric often surfaces, as this distribution might be perceived as subjective and potentially not encapsulate actual significance or implications. The evolving nature of the BI component also necessitates that the meta-metrics be frequently recalibrated or redefined to maintain their relevance. Moreover, the accuracy of individual metrics plays a pivotal role; any miscalculation in these foundational metrics can misguide the interpretation of the meta-metric. Finally, without a deep-rooted understanding of the constituent metrics, stakeholders could misconstrue the ramifications of a specific metameric value.

Despite the shortcomings mentioned before, the benefits of employing meta-metrics are manifold. The streamlined nature of meta-metrics ensures strict reporting, allowing BI managers to present consolidated insights to leadership, aiding in a more efficient grasp of the system's health and performance. They also pave the way for a standardized modus operandi for evaluating BI components across diverse departments, initiatives, or even organizations. The act of aggregating related metrics might unveil correlations or insights that would be otherwise concealed when examining metrics in isolation. This holistic focus, inherent to meta-metrics, facilitates BI teams to strategize comprehensive improvements as opposed to merely addressing individual anomalies. Furthermore, the very essence of meta-metrics simplifies the communicative process, which is especially beneficial when elucidating intricate BI concepts to non-technical stakeholders.

5. Quality Validation Using Metrics and Metametrics

5.1. Metrics within the V&V Process

The metrics mentioned in the previous section may play a positive role during the Validation and Verification (V&V) phase of an IS, presenting objective and quantifiable goals to evaluate system efficacy, performance, and alignment with predetermined standards. Through the application of these metrics, stakeholders can attain an augmented assurance of the system's deployment readiness and its aptitude to cater to both user and business imperatives.

V&V can be described as a process used to discover software defects and to confirm that the software is of quality (to some extent) in relation to some of its features. It is used to detect errors but also to evaluate quality factors such as reliability, security, usability, etc. Software testing is implemented through test scenarios that employ metrics. The modern approach to designing test scenarios is to generally describe the initial state of the user, the steps to be followed in order to test the software (after setting the appropriate goal for the test), and finally, to describe the expected outcome. Metrics play a pivotal role in defining the goal of test scenarios, how they will be implemented and the success threshold. This process offers flexibility. The basic principles of software testing through test scenarios follow a list of principles:

- Principle 1: The purpose of testing is to discover errors and assess the quality of the software.
- Principle 2: A good test scenario is likely to discover a new error or a quality failure (corresponding metric success value not obtained).
- Principle 3: The results of a test should be meticulously reviewed.
- Principle 4: A test scenario must necessarily include the expected output data. Desired output results are essential but are often overlooked. For software that performs simple operations, the tester calculates the expected values without having previously budgeted them. This is difficult to happen in specialized software or in pieces of software that interact with each other. Metrics facilitate the measurement in such complex cases.
- Principle 5: Test scenarios should be designed for both valid and invalid input data.
- Principle 6: The likelihood of defects in the software is proportional to the number of defects that have already been identified. The greater the number of errors detected in a piece of software, the greater the likelihood that there are others that have not been discovered. This principle is based on the empirical observation that errors occur in clusters.

Metrics for various components, such as BI, E-commerce, and more, offer objective standards that can be effectively harnessed to streamline and enhance the V&V processes. For instance, metrics related to the completeness and correctness of transactions, be it in E-commerce or BI queries, furnish verifiable evidence to confirm that specific functions of the system are operational as envisioned. In particular, metrics like “Number of complete BI queries” can be instrumental in verifying the proficiency of the BI component in data retrieval. Additionally, increasing performance efficiency is pivotal. Measures like page load times, BI dashboard load times, and throughput can serve as tangible indicators during validation. If the system falls short of these benchmarks, optimization might become imperative before it’s ready for deployment. The intersection of components and their interdependence can be gauged using meta-metrics, offering a panoramic view of component interactions. For example, the “Transaction Efficiency Index” meta-metric can corroborate that the BI and E-commerce segments of the system synergistically bolster each other’s efficiency.

5.2. Thresholds and Benchmarking

Establishing corresponding thresholds or benchmark scores for meta-metrics is pivotal for the practical assessment of complex IS. These benchmarks, which act as reference points, provide a context within which the computed values of meta-metrics can be evaluated to determine the system's performance relative to predetermined standards.

One approach to determining these benchmarks is through the analysis of historical data. If an organization maintains a repository of data accrued over extended periods, this data can be analyzed to discern typical value ranges and averages for meta-metrics. Specifically, historical records can be scrutinized to determine significant percentiles, with the median often serving as a central reference point and other percentiles indicating variations.

Moreover, competitive benchmarking presents another robust methodology. In this context, an organization's BI system meta-metrics are juxtaposed with those of industry peers or accepted industry benchmarks. By making such comparisons, the organization can ascertain its position in relation to its competitors or industry norms. It's the pursuit of meeting or surpassing these industry standards that can guide the establishment of internal benchmarks.

Furthermore, eliciting the judgment of domain experts can provide invaluable insights. Experts, due to their deep understanding of IS and their intricacies, are equipped to offer insights into the criteria that might classify performance as satisfactory, commendable, or exceptional. This can be achieved through organized expert panels, comprehensive interviews, or using techniques like the Delphi method to arrive at a consensus regarding appropriate benchmarks.

An organization might also consider a goal-oriented strategy, whereby performance objectives are explicitly set for the IS. Once articulated, these goals can be reverse-engineered into quantifiable benchmarks for the meta-metrics. The focus here is on ensuring that the metrics align with the broader objectives of the IS and, by extension, the organization.

Statistical methodologies also offer a way to discern benchmark values. For instance, adopting the Six Sigma philosophy might involve defining thresholds that are, say, three standard deviations from the mean, encompassing the majority of the data points and deeming any deviation from this as noteworthy or anomalous.

It is also worthwhile to consider user feedback in this context. End-users, being the primary beneficiaries of the IS, are often in an optimal position to provide feedback on its performance. Their perspectives can be collated and subsequently translated into quantifiable benchmarks for pertinent meta-metrics.

Given the dynamic and evolving nature of IS, it becomes imperative to emphasize iterative refinement. As systems change and as more data is collated, there arises a necessity to continually revisit and recalibrate these benchmarks. This iterative process ensures that benchmarks remain germane and are reflective of the current realities of the system. Lastly, a composite approach often yields optimal outcomes. A synthesis of insights from historical data, expert feedback, and competitive benchmarking can be integrated to derive comprehensive and robust benchmarks. Such a holistic approach ensures that the benchmarks are both grounded in empirical evidence and strategically aligned with the organization's vision.

The benchmarking process is another way of obtaining metric boundaries; its intricacies and variations often demand a nuanced appreciation of both the technical and contextual aspects surrounding BI systems. While technical data provides the foundation for metrics and benchmarks, the broader business context within which an IS operates can offer valuable insights into how these benchmarks should be interpreted and applied.

When employing historical data and statistical methodologies, it's imperative to account for changing business landscapes and technological advancements. What was deemed an acceptable benchmark a few years ago may no longer hold relevance in the face of recent innovations and shifts in industry standards? This underlines the importance of ensuring that benchmarks are not just historically grounded but are also forward-looking, taking into consideration projected trends and anticipated evolutions in the IS domain.

Similarly, while competitive benchmarking offers a relative perspective on performance, it's crucial to understand the unique challenges and opportunities that an individual organization might face. Blindly emulating industry standards without accounting for specific organizational contexts might lead to misaligned priorities and strategies.

Expert judgment, while invaluable, brings with it the need for critical evaluation. Experts, despite their depth of knowledge, come with their biases and perspectives. Therefore, a diverse panel of experts is often recommended to ensure a comprehensive and balanced view. Engaging experts from various domains—such as data science, business strategy, and user experience—can provide a multi-faceted perspective on setting benchmarks.

The emphasis on user feedback, meanwhile, underlines the growing importance of user-centricity in modern IS. As these systems increasingly cater to non-technical stakeholders, ensuring that benchmarks resonate with user expectations and experiences is paramount. It's not just about how efficiently a system processes data but also about how effectively it communicates insights to its users.

Furthermore, the iterative nature of benchmark refinement speaks to the continuous improvement paradigm inherent in effective IS practices. As organizations grow, their data needs evolve, and BI systems must adapt in tandem. Regularly revisiting benchmarks ensures alignment with current organizational objectives and paves the way for sustained excellence.

6. Conclusions

Using metrics for designing new complex IS or upgrading existing ones brings about a host of advantages. Firstly, metrics provide an objective and quantifiable means to evaluate various facets of the system, from functionality and performance to security and maintainability. This quantification enables designers and developers to make informed decisions based on quantitative and qualitative evidence rather than intuition or experience alone. By emphasizing evidence-based decision-making, metrics can guide optimization efforts, helping teams prioritize areas that offer the most significant returns in system improvements.

Moreover, metrics foster accountability and transparency in development processes. By setting clear, measurable targets, teams can maintain a sharp focus on critical requirements and quality standards. Such a structured approach also simplifies communication among stakeholders, as it provides a common language for discussing system attributes and performance. Metrics further facilitate continuous monitoring, enabling timely identification and mitigation of issues before they escalate, thus ensuring the robustness and resilience of the IS.

However, the use of metrics isn't without its limitations. One prominent challenge is the potential for over-reliance on quantifiable measures at the expense of qualitative insights. Not all vital aspects of a system, especially those concerning user experience or innovative features, can be easily quantified. There's also the danger of falling into the trap of "measurement for measurement's sake," where the sheer volume of collected metrics can overshadow their actual utility.

Furthermore, metrics may sometimes provide a narrow or myopic view of system performance, neglecting broader systemic issues or interdependencies. There's also the risk of prioritizing metrics-driven performance over more intangible yet critical aspects like user satisfaction, trust, or ethical considerations.

Looking towards the future, research in this arena could emphasize the development of more holistic and integrative metrics that encapsulate not just isolated system attributes but the broader ecosystem in which the IS operates. There's growing recognition of the need for metrics that can capture the ethical, social, and environmental implications of IS designs. Additionally, as systems grow more complex and intertwined with sociocultural contexts, interdisciplinary research involving sociologists, anthropologists, and ethicists could provide richer, more nuanced metrics. The integration of artificial intelligence could also play a pivotal role, offering dynamic metrics that evolve based on system performance, user feedback, and environmental changes. This dynamism could pave the way for adaptive IS designs that proactively evolve based on continuous feedback loops, driving the next frontier in complex information systems development.

Advancing further into the potential of metrics in IS design and upgrade, there's an evident interplay between technological advancement and metrics evolution. As technologies such as edge computing, quantum computing, and the Internet of Things (IoT) become more mainstream, they'll undoubtedly necessitate the development of new metrics tailored to their unique challenges and opportunities.

The ever-increasing focus on user-centric design and personalization in systems also underscores the need for more user-oriented metrics. These metrics would not just gauge system performance in isolation but also its efficacy in meeting diverse user needs and preferences. As systems grow increasingly adaptive, metrics that measure the system's ability to learn and evolve based on user behavior will gain prominence.

The use of metrics in complex IS design and upgrade are geared towards more integrative, adaptive, and user-centric paradigms. The synergy of technological advancements, evolving user needs, and the imperative for ethical considerations will shape the future discourse and innovation in this domain. Embracing these shifts and anticipating future challenges and opportunities will be instrumental for researchers, designers, and practitioners aiming to push the boundaries of what's possible in IS design and functionality.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Santos, D.S., Oliveira, B.R.N., Kazman, R. and Nakagawa, E.Y. (2023) Evaluation of Systems-of-Systems Software Architectures: State of the Art and Future Perspectives. *ACM Computing Surveys*, **55**, Article No. 67. https://doi.org/10.1145/3519020

[2] Ndukwe, I., Licorish, S, Tahir, A. and MacDonell, S. (2023) How Have Views on Software Quality Differed over Time? Research and Practice Viewpoints. *Journal of Systems and Software*, **195**, 111524. https://doi.org/10.1016/j.jss.2022.111524

[3] Simões, R., Melo, G., Abreu, F.B. and Oliveira, T. (2021) Towards Understanding Quality-Related Characteristics in Knowledge-Intensive Processes—A Systematic Literature Review. *Quality of Information and Communications Technology*, 197-207. https://doi.org/10.1007/978-3-030-85347-1_15

[4] Russo, D., Ciancarini, P., Falasconi, T. and Tomasi, M. (2018) A Meta-Model for Information Systems Quality: A Mixed Study of the Financial Sector. *ACM Transactions on Management Information Systems*, **9**, Article No. 11. https://doi.org/10.1145/3230713

[5] Nistala, P., Nori, V.K. and Reddy, R. (2019) Software Quality Models: A Systematic Mapping Study. *Proceedings of the International Conference on Software and System Processes*, Montreal, 25 May 2019, 125-134. https://doi.org/10.1109/ICSSP.2019.00025

[6] Stefani, A. (2022) Mining Metrics for Enhancing E-Commerce Systems User Experience. *Intelligent Information Management*, **14**, 25-51. https://doi.org/10.4236/iim.2022.141003

[7] Stefani, A. (2020) A Metrics Ecosystem for Designing Quality e-Commerce Systems. *International Journal of Computer Science & Information Technology* (*IJCSIT*), **10**. https://doi.org/10.2139/ssrn.3615445

[8] International Standardization Organisation (2014) ISO25000: Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE. ISO.

[9] Siebert, J., Joeckel, L., Heidrich, J., Trendowicz, A., Nakamichi, K., Ohashi, K., Namba,

I., Yamamoto, R. and Aoyama, M. (2021) Construction of a Quality Model for Machine Learning Systems. *Software Quality Journal*, **30**, 307-335. https://doi.org/10.1007/s11219-021-09557-y

[10] International Standardization Organisation (2011) ISO25010: Software Engineering—Software Product Quality Requirements and Evaluation (SQuaRE)—Measurement Reference Model and Guide. ISO.

[11] International Standardization Organisation (2008) ISO25012: Software Engineering—Software Product Quality Requirements and Evaluation (SQuaRE)—Data Quality Model. ISO.

[12] Fadlallah, H., Kilany, R., Dhayne, H., El Haddad, R., Haque, R., Taher, Y. and Jaber, A. (2023) BIGQA: Declarative Big Data Quality Assessment. *Journal of Data and Information Quality*, **15**, Article No. 27. https://doi.org/10.1145/3603706

[13] Dietmar Winkler, D., Biffl, S. and Bergsmann, J. (Eds.) (2019) Software Quality: The Complexity and Challenges of Software Engineering and Software Quality in the Cloud. 11*th International Conference*, *SWQD* 2019, Vienna, 15-18 January 2019, Springer. https://doi.org/10.1007/978-3-030-05767-1