JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Automated Data Quality Assessment And Enhancement For Saas Based Data Applications

Saransh Arora,Sunil Raj Thota,

Independent Researcher/Sr. Data Engineer, AI/ML & Data Engineering[1]
Independent Researcher/Sr. Software Engineer, AI/ML & Full-stack Web Dev[2]

Abstract: Since the early days of computing, data quality issues have been prevalent, leading to the development of various domain-specific techniques to assess and improve data quality, particularly in relational databases and data warehouses. However, with the emergence of big data analytics and the resurgence of machine learning, there is a need to reevaluate the suitability of traditional relational database-centric approaches to address data quality in these new contexts. In this paper, explore the nature of data quality issues in the realm of big data and machine learning. This paper discusses different facets of data quality and proposes a data governance-driven framework for managing the data quality lifecycle in this evolving landscape. Additionally, we outline an approach to implementing this framework effectively. We also present a summary of the available tools for managing data quality and explore potential future developments in this area. Our focus is on automated data quality assessment and enhancement, particularly in software-as-a Service (SaaS)-based data applications, where ensuring data quality is crucial for reliable insights and decision-making.

Keywords - Software as a Service, Data Quality. SaaS quality

I INTRODUCTION

In today's data-driven landscape, businesses heavily rely on software-as-a Service (SaaS) applications to manage and process vast amounts of data. These applications, spanning various domains such as customer relationship management (CRM), enterprise resource planning (ERP), and business intelligence (BI), are essential for making informed decisions, optimizing operations, and enhancing customer experiences. However, data quality determines these applications' efficacy. Poor data quality can lead to incorrect insights, suboptimal decisions, and operational inefficiencies, ultimately affecting an organization's bottom line. Software maintenance comprises a significant portion (70%) of the total software implementation costs [1-3] state that maintaining and running existing systems and software infrastructure consumes more than 75% of the IT budget. The increase in development and operating costs, which was also one of the main reasons for the failure of application service providers (ASPs) in the 1990s (De Miranda, 2010), is inevitable. As a result, the demand for a software as a service (SaaS) model is increasing because the costs of hardware, technology, maintenance, and tenant management are lower [4] for the implementation of SaaS applications.

Customization is an essential requirement for providing the same application to different users [2], as they may have different business flow, interfaces, and data [5] As a result, this requirement will pose quality challenges and risks for the SaaS application hosts.[3] User-specific customization influences all SaaS application components, including both functional and non-functional aspects of all layers of SaaS architecture.

Another complication has to span multiple layers of SaaS architecture [4]. All SaaS application elements, including those with cross-layer relationships, must be customizable. Moreover, customization includes adjustments to the software’s source code, which becomes highly complex in the SaaS model. [5]

Changes in the requirements often occur after applications and services have been developed; therefore, runtime customization must be provided within the same software instance for different users and should not impact their privacy or the application's availability [6].

Generally, SaaS applications lack the customizability of on-premises applications, which would result in reduced software maintenance. By contrast, frequent customization of the SaaS application would require a burdensome maintenance process and pose a significant challenge to scalability and cost-efficiency. Therefore, application vendors should be cautious about their technical capacity when making customization assessments, especially when customization impacts the crucial features of SaaS [9].

There is insufficient evidence in the available studies to assess the effect of software customization on SaaS attributes .Therefore, it is crucial to specify the type of customization to evaluate the associated impact and risk, as any alteration is likely to impact the software quality. Despite the need for several researchers to consider SaaS application customization, there has been no clear effort to categorize software customization types and practices in a multi-tenant context.

Accordingly, research is required to establish a clear model that considers: (1) generic software customization types and a list of common practices for each client in the SaaS multi-tenant context; and (2) key quality attributes associated with customization. This study provides detailed evidence of the content validity and reliability of the proposed model. For content validity, two main calculations are considered: the item content validity index (I-CVI) of each customization practice and SaaS quality attributes, as well as the scale content validity index (S-CVI/Ave). In this study, we evaluate two quantities to determine the internal consistency and reliability of the model: Cronbach's alpha coefficient and the corrected item-total correlation.

The structure of this manuscript is as follows: The next section discusses the related works. The third section presents the model's conceptualisation. The fourth section explains the methodology used, whereas the fifth section reports the results of the conducted study, followed by a discussion in the sixth section and threats to validity in the seventh section. The eighth section finally presents the conclusions and future work.

Software quality has the objective of ensuring conformity with software requirements. It means that a software product must meet both explicit and implicit specifications, and it is offered as a package or a service (SaaS, Software as a Service). Therefore, the final goal is to ensure a product that satisfies all clients' expectations [1].

Software development companies are increasingly concerned about the quality and excellence of their products in response to the growing demands of organizations. Nowadays, software products are critical tools, and their quality requirements have become key criteria in software acquisition [2,3]. It directly reflects the mission and objectives of an organization's business.

In this context, software as a service has gained popularity in organizations. The success of those products depends on functional and non-functional requirements [4]. These attributes help define the quality of a software product. However, the complexity of measurable requirements for physical products is linked to the difficulty in establishing and using quality criteria for software products [5].

SaaS is a new paradigm based on software delivery, in which client organizations pay not for the ownership but for the use of software. As a result, software companies have to maintain and evolve software, as well as assist with technical support issues. Many software companies are already applying this new paradigm because of the market's concurrence. However, these companies face challenges primarily in ensuring the quality of their final products, given that SaaS is a relatively new and emerging paradigm [6].

Only a few studies look at the definitions and impacts of quality attributes related to software-as-a-service products. In consequence, there's no evidence in scientific research about methods, models, and metrics that provide a practical and efficient assessment of quality in SaaS products.Many methods for assessing the quality of software products are ineffective when compared to the SaaS paradigm, primarily due to their inability to provide a precise set of quality attributes. An efficient attribute set should include features and sub-features that demonstrate its scope and applicability. Moreover, it needs to show its positive and negative impacts in front of applications [6, 7].

II RELATED WORK

Ali et al., 2019b initially constructed a software customization model for SaaS quality, and this study presents an iterative approach to develop, refine, and improve it. The main components of this model are the customization types, common customization practices of each type, and quality attributes of SaaS applications associated with customization. To our knowledge, no one has developed and validated a model based on these criteria. However, in this section, we review the literature on generic SaaS customization options, followed by the literature on quality models for SaaS applications.

SaaS customization

Scholars [7] and [8] have suggested different types of customization based on the layers of SaaS architecture and customization objects. [9] illustrated five types of customization: GUI customization, service customization, process customization, data customization, and cross-layer customization. Tsai & Sun (2013) considered the customization of GUI, service, process, data, and QoS. [10] defined three different types of SaaS customization: user interface, workflow, and access control.

On the other hand, some studies categorized SaaS customization according to the methods used. [11] Explained three types of customization: source code, composition, and configuration. [12] Proposed three types of customization for multi-tenant SaaS applications: desktop integration, user-interface customization, and back-end customization, based on the hosting and execution locations of the customizations.

Moreover, identified the types of customization in a tenant base. Segment variability and tenant-oriented variability are the two classifications for customization; the former performs customization based on the requirements of a tenant community, while the latter performs customization based on the specific requirements of a tenant.

SaaS quality

Many studies have focused entirely on defining and identifying the quality attributes of SaaS applications. For instance, [14] proposed a list of 33 quality attributes for SaaS and provided their definitions, and [15] proposed a comprehensive quality model for assessing SaaS cloud services. These authors identified characteristics and quality attributes using ISO/IEC 9126 [17] and defined metrics to measure them. La & Kim (2009) proposed a systematic process to build a high-quality SaaS application, considering the main SaaS design criteria.

[16] proposed a ‘‘SaaS Quality" method for evaluating the quality of SaaS applications. The proposed method generated the SaaS quality model, based on ISO/IEC 9126 [18] and IT management models (Publications Service Management, 2008; IT Governance Institute, 2007). Another related study extracted the critical quality attributes for SaaS reusability and identified SaaS reusability metrics for every quality attribute [19-20] have customized software quality models to fit the SaaS context, classifying the SaaS quality criteria for products and processes and identifying quality attributes for each class.

III RESEARCH METHODOLOGY

This research, which focuses on objective exploratory research using case study procedures as well as documental and bibliographic research, falls under the qualitative and technological categories [17,18]. This work adheres to a four-phase methodology.

Conception and preparation of research: The first phase encompasses the conceptualization and preliminary stages of implementing the research. The definition of the problem considers the requirements of the research environment, ensuring the validity and relevance of the efforts in light of the absence of a practical and efficient method for assessing the quality of a SaaS product.

Preliminar results and analyses: This phase involves the realization of analyses and studies related to the characteristics and quality requirements of the SaaS paradigm, which enables the construction of a specific quality model for SaaS. The initial phase conducted an analysis of software quality requirements, encompassing studies of norms and standards, and offering a review of quality criteria typically found in conventional software products. This analysis made it possible to identify which characteristics have applicability and purpose within SaaS systems. Simultaneously, we conducted an analysis of requirements concerning the quality of IT services. We used frameworks and guides to good practice in IT service management to raise and analyze quality criteria, which also apply to software as a service (SaaS), bearing in mind that SaaS shares similar characteristics with conventional systems and IT services [4].

Following the analysis, we conducted a survey on quality requirements within the SaaS context. We objectively identified and combined these criteria and requirements into a single set for a SaaS product, thereby facilitating their application in an evaluation process. We integrated the criteria using comparisons and definitions of applicability based on the standards ISO/IEC 9126 [19], ITIL v3 [20], and COBIT 4.1 [21], with the aim of identifying quality characteristics more relevant to the SaaS context. After the identification and integration of the criteria, there has been a complementation and prioritization of these characteristics by means of a search of 85 people related to the SaaS paradigm (entrepreneurs, experts, and researchers), offering greater complementation and reliability to the characteristics that have been raised.

Validation and improvements in the method: This stage refers to the initial phase of the case study, with the primary objective being the validation of the proposed method and the quality model. At this stage, a process of evaluation has been defined, as has a stage of continued improvement for SaaS quality. Once SaaSQuality defined the quality characteristics, it became necessary to convert these characteristics into quantifiable attributes for measurement and scoring. This mapping has facilitated the preparation of a list of verifications (checklist), enabling a practical assessment of the quality requirements.

For the practical application of the method, there was a need to set up an evaluation process that could be repeatable, reproducible, impartial, and objective. We based the SaaS quality evaluation process on the standard ISO/IEC 14598 [22], but made some changes and adjustments to better align with the SaaS products and the roles of the process actors.

At the conclusion of this phase, we have implemented validation and enhancements in SaaSQuality through a case study. The case study is important because it shows the objective achieved, limitations, and possible improvements in a real environment, seeking to validate the work performed. To conduct the case study, we evaluated the product "Sales Force CRM Sales" using the guidelines and metrics of the SaaS Quality method.

At this stage, we have achieved results through a comparison, in addition to examining the partial results we've obtained and published in the scientific community [23], particularly in the field of "Quality of Software." We compared the SaaSQuality method with a conventional quality evaluation method, MEDE-Pros, to enhance its validation. This comparison revealed several points that conventional methods do not discuss or assess, along with some limitations.

Quality Model for SaaS Quality

According to the authors [10], for a method of evaluation to be efficient, it must be based on or incorporate a quality model specific to the context of its application. The authors of this work did not find any models of quality, either in the literature or in international standards, that adequately capture the characteristics of a SaaS system. In light of this particular characteristic, it was necessary to propose and define a model that contemplates criteria and quality requirements specific to the SaaS.

We propose a model that incorporates various characteristics from the ISO 9126 standard, ITIL v3, and the COBIT 4.1 framework, providing insights into conventional software products and IT services. The standard ISO 9126 has provided well-defined characteristics of a software product, stating aspects of functionality, reliability, usability, efficiency, maintainability, and portability. For its part, ITIL v3 and COBIT 4.1 were important elements for the identification of characteristics that led to the improvement of the management and quality of services for TI, particularly in relation to legal issues, performance, and security in a SaaS product.

In addition to the support of those methodologies, the authors also drew from other works [8, 9, 10], which included various authors' perspectives on the quality of SaaS systems. The lack of practical application necessitated prioritizing and enhancing the highlighted characteristics of this model.

We conducted a search online with various experts in the field, including participants, managers, and consultants from all areas of SaaS, to provide support. The online search was based on sending a checklist to the experts, who were able to analyze and indicate the relevance of each requirement raised. Based on the specialists' knowledge and practical experience, the relevance of the quality requirements has revealed which characteristics are truly important. We can define the relevance based on some criteria of the trial [26].

Very Relevant: Important features are crucial for ensuring their quality, as they directly impact both the internal and external quality of SaaS products.

Relevant features primarily pertain to the external quality of the system, which significantly influences the final quality of the product.

Little Relevant: Certain characteristics are desirable but not essential to the application's success and operation.

Not applicable: features that don't apply to the SaaS context. There are aspects of quality that are applicable and measurable in conventional software but not in the context of SaaS, such as, for example, installation.

We have defined a hierarchical structure based on the highlighted characteristics to aid in the implementation and upkeep of the proposed model and enhance its comprehension. The hierarchical structure defined in this work bears a resemblance to the structure proposed by Hoffman et al. [27], which aims to simplify the verification of quality requirements during the construction of a SaaS product. For the current work, there have been three levels of hierarchy, each with a specific purpose. We present the three levels of this structure in the following order:

Level 1 - Criteria for Evaluation: Have been defined criteria to be observed during the evaluation of the quality in a SaaS system;

Level 2 - Requirements (Guidelines): Formed by requirements of quality, representing guidelines specific for each criterion defined in the first level;

Level 3 - Attributes of Verification: The third level is formed by the attributes of verification, in other words, measurable attributes that can be measured and punctuated.

In this way, the graphical representation of the model of quality proposed for the method "SaasQuality" was determined in accordance with the Figure 1.

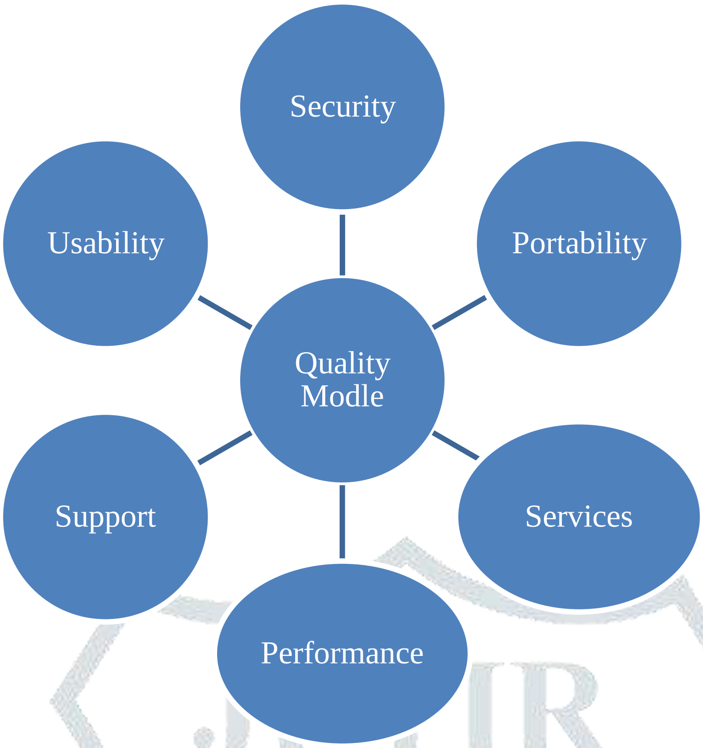

Figure 1 - Set of criteria and quality requirements specific to SaaS.

The following is presented with a summary of the main impact that the proposed requirements present in relation to SaaS systems.

Functionality: Through the use of architectures and technologies, a SaaS system provides high interoperability between systems, together with the efficient adequacy of their functions.

Usability: A system with good usability can reduce the services within the organization's support, allowing for better human interaction with the system and preventing performance issues.

Security: The need for cryptography, integrity, and confidence in a SaaS approach requires attention to detail in its architecture. The development of many SOA architecture standards to support safety is still in its infancy [28].

Performance: A SaaS approach may have a negative impact on application performance due to network delays and user burden on the same system. The contracting organization must design and assess the quantity of users that will use the system. The supplier must design and carefully assess their services by making sure that the performance requirements needed will be met.

Support: The ability to support in SaaS systems can provide a negative effect on the contracting company and provider. Employing a large number of outsourced services can negatively impact testability.

Service Level: The contracting parties of SaaS systems can negotiate a SLA with their supplier by providing agreements on levels of availability and sanctions for non-compliance with the agreement. If the system does not incorporate the necessary audit skills, auditability may suffer.

When managing adaptations appropriately, using a SaaS approach should enhance portability, with the main concern being the portability of web browsers rather than operational systems.

Presentation of Saas quality

We proposed and modeled SaaSQuality based on the logic of other evaluation methods [29, 30, 31]. All referenced works follow a generic structure, generally relying on a specific model of quality along with a well-structured evaluation process. Figure 2 shows an overview of SaaS quality.

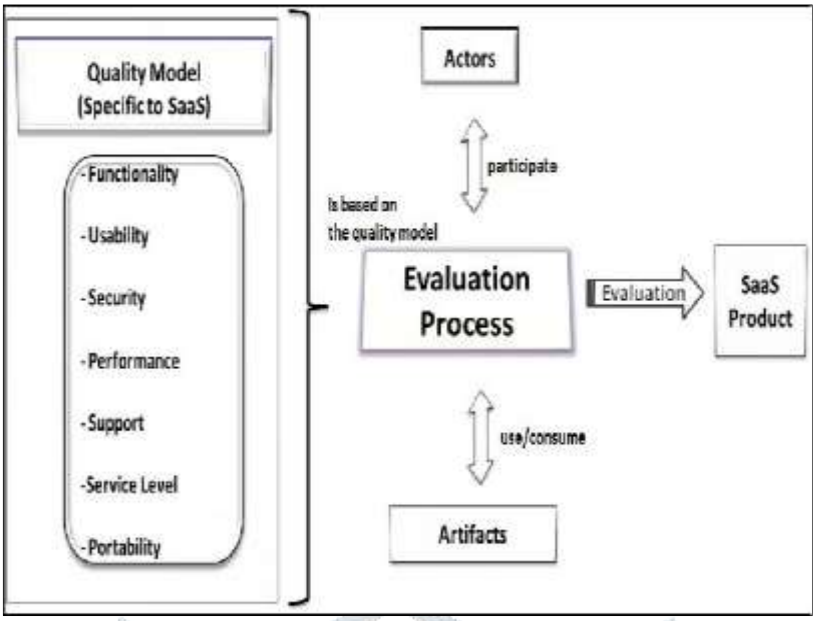

Figure 2 - Overview of SaaSQuality

For the subjectivity of the evaluation is minimal, ensuring greater efficiency, the process of SaaSQuality was developed following certain characteristics [4]:

Repeatable: The process should make it possible to apply the method multiple times in various contexts and still produce results that are either identical or similar.

Reproducible: The process should include specifications and steps so that various evaluators (different entities) can apply the method and produce results that are acceptable in the evaluation context.

Impartial: In relation to practical evaluation, the process should avoid assuming that internal or external factors influence the results.

Objective: The process should ensure clear results, avoiding ambiguities and different interpretations.

In the context of the current work, the actors represent roles that a human being plays in relation to the activities of the evaluation process. Basically, there are four actors who participate in this process.

Requester: A person or entity who requests an assessment of a specific SaaS product, clearly stating their goals and objectives, is required.

Evaluator: The individual in charge oversees all the guidelines for implementing the SaaSQuality method, acting as a facilitator and liaison between the process and its participants.

Users Interviewed: Participants that have the purpose of giving critical views (positive and negative) in relation to quality aspects evaluated within the SaaS product;

Specialist: A person who possesses comprehensive knowledge about the context of the system under evaluation, specifically the SaaS, is in question.

The actors performed activities that were present in stages of the evaluation process. The steps outlined in Figure 3, which are specific to SaaS quality, are based on best practices and expertise required in the process of evaluating ISO/IEC 14598. However, they simplify and focus on systems SaaS. Below is a brief summary for each stage of the process.

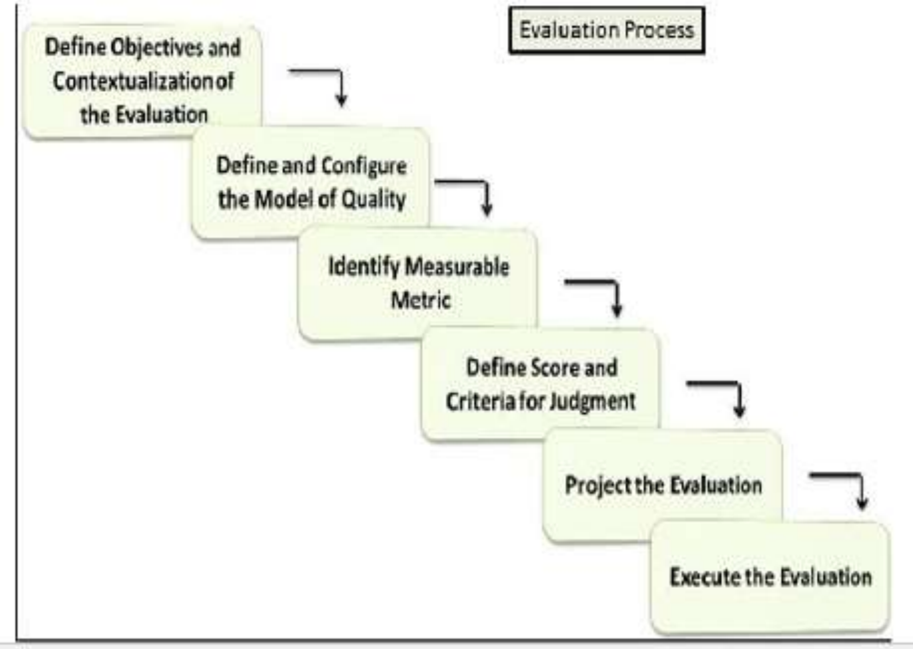
Evaluation Process
Define Objectives and Contextualization of the Evaluation
Define and Configure the Model of Quality
Identify Measurable Metric
Define Score and Criteria for Judgment
Project the Evaluation
Execute the Evaluation

Figure 3 - Stages of the evaluation process.

Define Objectives and Contextualization of the Evaluation: This phase aims to clearly define the purpose of the upcoming evaluation. The requestor will specify the software for evaluation, including its version and other necessary requirements.

Define and Configure the Model of Quality: Once the practical assessments have defined the model of quality, the next step is to configure it, indicating the criteria and quality requirements it will employ.

Identify Measurable Metric: This stage involves mapping the characteristics of quality into measurable attributes, enabling their scoring and evaluation.

Define Score and Criteria for Judgment: In this stage, the numerical values for the measurable attributes will be set, establishing criteria for judgment for the comparisons of the results obtained.

Project the Evaluation: In this stage, the evaluation plan to be followed during the practical assessments should be defined, containing guidelines and procedures that allow an application to be structured and repeatable.

Execute the Evaluation: The evaluator must use all of the information from the earlier stages in the final stage of the evaluation process, which leads to the following activities: gathering measures, comparing them to criteria, and evaluating the results.

Evaluations may result in minor changes to some of the presented stages; more specifically, the evaluator and specialist will make these changes.

IV RESULTS OF VALIDATION AND EVALUATION

The research underwent validation in two stages. We first applied it to a case study involving a SaaS product to validate the proposed method, and then conducted a comparative analysis between the SaaS Quality and the MEDE-Pros to identify similarities, differences, and limitations between both methods.

Case Study ("Salesforce CRM Sales")

The case study was carried out by applying the SaaSQuality in relation to the product "Salesforce CRM Sales", a Customer Relationship Management (CRM) system that has the purpose facilitate and improve the relationship of the organizations and their respective customers. The following will be done the detailing of each step of the process of evaluation, relating them with the case study, identifying what was done at each stage of the process:

Define Objectives and Contextualization of the Evaluation: In the context of this work, the objective of the application of the method for the study of the case was to evaluate the SaaSQuality, in relation to your model of quality, process of evaluation and the techniques used. The expectation of the study is also assess the product “Sales force CRM Sales”, noting their compliances with requirements of quality.

Define and Configure the Model of Quality: We adopted the quality model from the section "Quality Model for SaaS Quality" (Figure 4) for our practical assessments. We chose and utilized this model not only for its inclusion of various criteria for evaluating the quality of a SaaS system, but also for its ability to divide requirements (guidelines) and evaluation attributes, thereby simplifying their application in practical assessments.

Identify Measurable Metric: At this stage, we separated the quality requirements into attributes that could be measured and punctuated. Auxiliary staff developed techniques and methodologies to support and facilitate the identification of measurable metrics. This evaluation employed a list of verifications as a technique to efficiently map the quality model's requirements into attributes and measurable items.

The specialist leveraged their knowledge to identify the measurable attributes in the quality model, formulating straightforward and objective questions that facilitate their assessment and organization into a checklist format. The specialist proposed a total of 70 issues for evaluation, breaking down the questions into seven distinct groups based on the quality model's criteria: functionality, usability, security, performance, support, service level, and portability. The list of verification proposals for the evaluations included additional information to enable the evaluator to use it as a template for application purposes. The evaluator has incorporated the additional information into the list of verifications, which includes details such as the average duration of the evaluations, the impartiality of the evaluation, the local definition, and the data collected, among other details, to streamline the execution and management of the evaluations..

Define Score and Criteria for Judgment: The verification list outlines 10 attributes in the form of questions for each quality criterion, resulting in a total of 70 items. We defined this number to facilitate the creation of a comprehensive list of verifications, without being exhaustive to avoid compromising the time required for the evaluations' implementation. We have defined the weight for each issue in the following way:

Maximum value: "1" (highest score);

Minimum value: "0" (lowest score).

We conducted 15 assessments for each quality criterion with a variety of individuals, resulting in a variety of assessment instances and notes regarding the final quality of each criterion. To facilitate a better comparison of the results, we calculated the arithmetic mean of the 15 evaluations, which resulted in 15 different values related to the quality of each criterion assessed.This process of getting the average quality was done for seven evaluation criteria, which separated the specific note of each criterion from the final note of the evaluated product. It is interesting for the person asking for the evaluation to be able to see the notes separately and check which quality criteria are below the required level of quality. This lets them do a trial quickly while making changes and improvements. The authors based their judgment on an analysis of around 300 software products [32]. The criteria are based essentially on three levels: superior, medium, and lower, and can be seen in Figure 4.

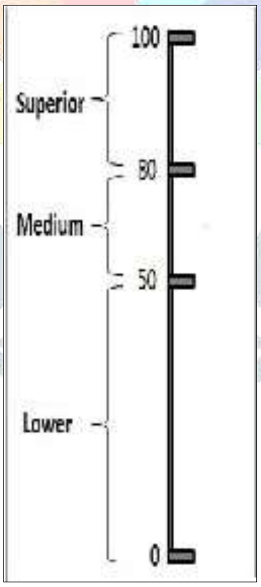

Figure 4 - Levels of quality of a product of software.

These criteria have been incorporated into the SaaS Quality, taking the following definitions and prospects of conclusion:

Superior level, The software product should meet at least 80% of the specified quality requirements. This means that the note can vary from 100 to 80 on a scale of 0 to 100.

Medium level, The software product should meet at least 50% of the specified quality requirements. This means that the note can vary from 80 to 50.

At a lower level, a software product can have a note of 50 to 0, indicating a low level of quality.

Project the Evaluation: The practice evaluations were conducted in accordance with specific guidelines. The users interviewed during the evaluations were students with advanced knowledge in information technology, with a certain familiarity with the SaaS product assessed.

We estimated and implemented the time of each evaluation between 40 and 50 minutes, aiming to stay within this range to ensure a swift assessment and implementation. Only the authors possessed all the information and data raised by the interviewed users, ensuring their confidentiality. Information obtained has no commercial value; it only adds value to this work's research and development. We have conducted a total of 15 assessments involving 15 distinct individuals. This number of assessments had objectives that improved the proposed method, as well as the convergence of the obtained data. Initial assessments revealed errors and details that needed improvement, affecting both the quality model and the evaluation process.

Execute the Evaluation: Basically at this stage of the case study the Evaluator has executed three tasks:

Collection of the measures

The process involves comparing the measures with pre-defined criteria before making a final judgment. The assessment is based on the information gathered during the evaluations. In accordance with activities of obtaining measures, all have been collected through the evaluations made by the users interviewed, making clear the impersonality of the evaluator that was present only to doubts and perform the evaluation in a repeatable manner. At the end of the evaluations and early of activity de comparison of criteria, the numerical data have been treated and compared with the criteria, enabling a final judgment. We have finished the process of assessing the data and created a comprehensive evaluation report. The goal of writing a final report is to highlight the positive and negative aspects of the quality criteria evaluated.

In addition to identifying relevant aspects of the SaaS product's quality, the report also identifies problematic aspects that occurred during the implementation of the evaluations. These include issues such as product unavailability, errors generated during the evaluation process, and a lack of information about the product, among others. Considerations and Results: At the end of all the evaluations, the product had an average note of 76.6 in relation to the final quality, which falls within the level of "medium" quality (Figure 5).

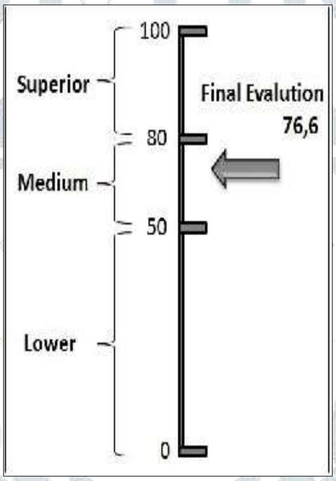

Figure 5 - The quality achieved by the "SalesForce CRM Sales"

Even by presenting a level "medium" of quality, not all the criteria of the evaluated product had notes unsatisfactory within the context SaaS (Figure 6).

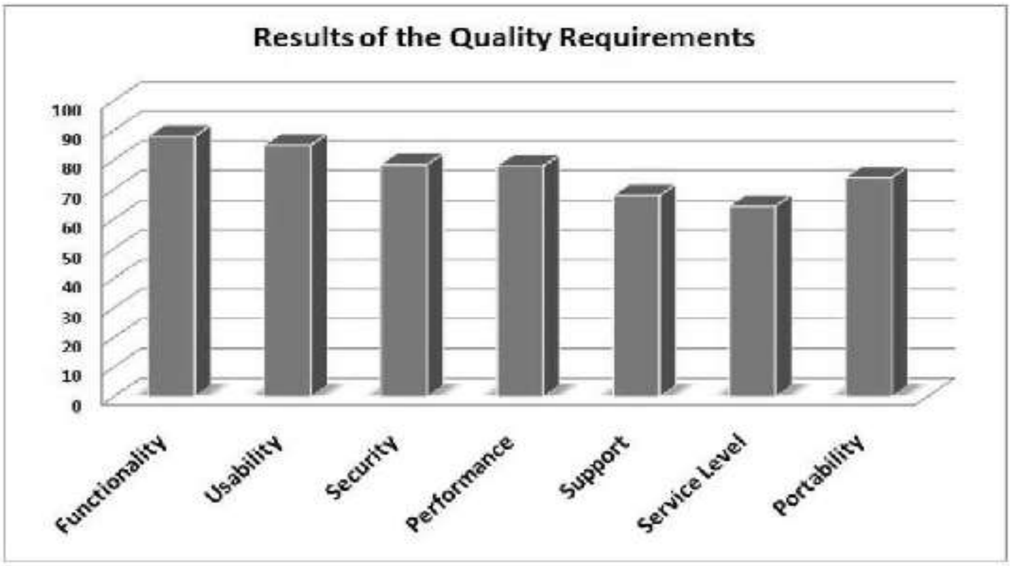

Figure 6 - Note assigned in relation to each criteria of quality.

Comparison with the method mede-pros

This comparison has been drawn up with the aim of responding to some questions that can be the case study has come to an end.

What is the need for developing a specific method for assessing SaaS products?

The conventional methods are not ready and appropriate to assess a SaaS product.

What are the similarities and differences in SaaS quality compared to conventional methods?

We conducted a comparative study between the methods SaaSQuality and MEDE, aiming to provide well-reasoned answers to the aforementioned questions.

Pros. In order to achieve this comparative some steps have been followed, allowing the comparison between the data and results of both methods.

Step 1: The first activity of comparative was the choice of a conventional method that will provide a practical assessment of the quality of a product of software, in the case of this research the MEDE-pros;

Step 2: The second stage, was the study and the detail of the model of quality in which the MEDE-pros is based, based on requirements and quality attributes;

Step 3: For the completion of the third stage, it was necessary to use a set of questions and checks (checklist) present in the book: "Quality of Product of Software" (Colombo & Guerra, 2009). This checklist was drawn up in order to be applied to the directly MEDE- pros;

Step 4: The fourth stage makes reference to the application of the method MEDE-pros on
the product "Salesforce CRM Sales";

Step 5: At this point, we compared the data from the MEDE-pros evaluations with the data from SaaSQuality's case study.

After analyzing and interpreting the data, we mounted a "radar"-style graphic (Figure 8) and plotted the values of each quality criterion along the axes, representing the two methods (SaaSQuality and MEDE-pros).

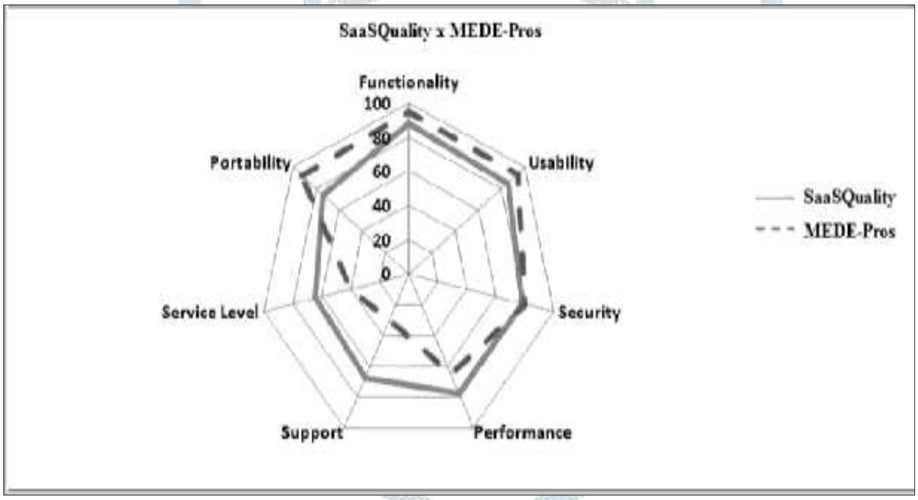

Figure 7 - The result of the comparative of quality criteria (SaaSQuality vs MEDE-pros)

Figure 7 illustrates the convergence and divergence of data in relation to quality criteria. Functionality, usability, and security have demonstrated a significant convergence of data in relation to methods. This is because conventional systems like SaaS share similar quality criteria. These similarities are briefly described below.

Functionality: It appears primarily in the areas of interoperability, access, and accuracy, thereby enabling the assurance of functions that accurately and correctly meet customer needs, as well as the potential for integration with other legacy systems within the organization.

Usability: In computing systems, the requirements of operability and attractiveness are necessary; they are extremely important to greater user satisfaction.

Security: Even though the paradigm of SaaS has a greater impact, security aspects are very important in conventional systems, especially those that work with critical business data, requiring greater reliability, integrity, and availability for these data.

The **Portability** The criterion also showed a convergence of data in both methods; however, this feature presents an application context that is quite different in bIn SaaS, the portability criterion primarily focuses on browsers and communication channels. unication. In conventional systems, the concept of portability is closely associated with operating systems and distributions. In other words, even if they are converging, the data obtained cannot be considered similar because the coexistence requirements and adaptability have very different applicability in both paradigms.

The criterion **Performance** There has been a slight discrepancy in the data, but this was anticipated as SaaS systems, with their tenant-based and multi-tenant access methods, offer unique features in terms of user scalability and the ability to add new services to the system [33, 34].

The **Service of Level** and **Support** have had the greatest divergence in the results. The reason lies in the direct influence of IT service management features not included in standard ISO 9126, like customizations, incident and problem management, continuity SLA plans, data management, and auditability, on the quality requirements. The following are the main points of the analysis of these requirements. The MEDE-pros method cannot measure the service level because it excludes features like Service Level Agreements (SLAs), data management, and continuity plans, among others, from its quality model. The support had a good relationship between the characteristics of testabil.The support demonstrated a strong correlation between the characteristics of testability and stability. However, when it comes to incidents and personalization, the characteristics of SaaS and conventional systems differ significant the comparisons conclude that a conventional method is not entirely suitable for evaluating SaaS products. The MEDE-pros method effectively assessed some characteristics, but the context of their use hindered the evaluation of others. This is due, among other things, to the use of the quality model.

V CONCLUSION

This paper proposes the development of a method for assessing the quality of Software as a Service (SaaS) product, known as SaaS Quality. The proposed method relies on a specific quality model for the SaaS paradigm, offering a straightforward and efficient evaluation process. The SaaS Quality team built a model that fits the SaaS paradigm and can be used in real life by combining criteria and principles from ways to manage IT services with criteria and principles from ways to evaluate the quality of software products. The criteria defined have enabled the lifting and impact of quality criteria specific to the SaaS context, dividing them into groups of characteristics and sub-characteristics, in order to show their impact, both positive and negative, on this new paradigm. The quality model is a compilation of the main features present in a software product or IT service. The evaluations performed by the method in question were carried out by means of a list of verifications (checklist), contemplating through measurable attributes the score and measurement of quality requirements. At the end of the research, the authors highlight important aspects that allow the extension of studies into future work. The initial point pertains to the suggestion of a technique for evaluating and ensuring quality throughout the SaaS product development process. The second approach focuses on measuring the SaaS quality in relation to its current state of use. During the problem definition phase, there has been a significant focus on both internal and external quaintly, the third step involves enhancing and expanding the list of verifications. The present paper places emphasis on attributes that offer a comprehensive evaluation of a SaaS product's quality, assessing each attribute equally and punctually.

References

1. Kan, S. H. (2002). Metrics and Models in Software Quality Engineering. Addison-Wesley.
2. Akella, P. S.; Rao, K. N. (2011) Effective Independent Quality Assessment Using IV&V. International Journal of Computer Science and Information Technology (IJCSIT), Vol 3 (3), 320-332.
3. Krishna, P.; Rao, R. (2011) Software Quality Assurance in CMM and XPA Comparative Study. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 2 (6) , 2915- 2919.
4. Lew, P.; Zhang, L.; Wang, S.; Jiang, W. (2008). Guidelines to Determine Quality for Web-Based Software Applications. In: Computational Intelligence for Modelling Control & Automation,p 709, IEEE Publications, Vienna, Austria.
5. Colombo, R. M. T.; Guerra A. C. (2009). Information technology: Quality of Product of Software, MCT/SEPIN.
6. Barney, S.; Wohlin, C. (2009). Software Product Quality: Ensuring a Common Goal. Lecture Notes in Computer Science 3(5), 256-267.
7. Auer, T. (1998). Quality of IS use. European Journal of Information Systems 7(9), 192-201.
8. Dubey, A.; Wagle, D. (2006). Delivering software as a service. The McKinsey Quarterly 3(5),1-12.
9. O’Brien, L.; Bass, L.; Merson, P. (2005). Quality attributes and service-oriented architectures. Technical report, Carnegie Mellon University.
10. Havelka, D.; Sutton, S. G.; Arnold, V. (1998). A methodology for developing measurement criteria for assurance services: An application in information systems assurance. Journal of Practice and

Theory 8(5),80-87.

11. Jureta, I. J.; Herssens, C.; Faulkner, S. (2009). A comprehensive quality model for service-oriented systems. Software Quality Control 7(4), 65-98.
12. Cancian, M. H. (2009). A proposal of reference guide for providers of software as a service. Dissertation in Automation and Systems Engineering, University Santa Catarina.
13. Cancian, M. H.; Rabelo, R.J.; Wangenheim, C. G. V. (2009). A proposal for drawing up of the contract for Level of Service to Software as a Service (SaaS). In 8th International Information and Telecommunication Technologies Symposium, p 410, IEEE Publications, Florianópolis, Brasil.
14. Rosenberg, F.; Leitner, P.; Michlmayr, A. P. C.; Dustdar, S. (2008). Towards composition as a service-a quality of service driven approach. Dstributed Systems Group.
15. Silva, E. M.; Yue, G. K.; Rotondaro, R. G.; Laurindo, F. J. B. (2006). Management of the quality of services for IT:in search of competitiveness. Magazine Production 2(6), 329-340.
16. Miguel, P. A. C.; Salomi, G. E. (2004). A review of the models for measurement of the quality of services. Magazine Production 10(1), 12-30.
17. Jung, C. F. (2004). Methodology for research and development: applied to new technologies. Axcel Books Publishing House.
18. Wohlin, C.; Runeson, P.; Host, M.; Ohlsson, M. C.; Regneel, B.; Wesslén, A. (2021). Experimentation in software engineering: An introduction. Kluwer Academic Plublishers.
19. ISO/IEC 9126. (2020). the international organization for standardization. Software engineering Product quality.
20. ITIL v3. (2018). Information technology infrastructure library. Publications Service Management.
21. TGI. (2017). Information technology governance institute. COBIT 4.1: Control objectives, Management guidelines, Maturity models. Rolling Meadows: ITGI.
22. ISO/IEC 14598-5. (2018). International standard. information technology - software product evaluation - part 5: Process for evaluators.
23. Duarte Filho, N. F.; Padua, C. I. P. S.; Zambalde, A. L.; Bermejo, P. H. S. (2010). Quality attributes for SaaS´s products. In: 5th Iberian Conference on Information Systems and Technology, p 80, IEEE Publications, Santiago de Compostela, Spain.
24. Boehm, B. W. (1981). Software engineering economics. Prentice-Hall.
25. Gatti, M. A. C.; Werneck, V. B. (2014). Quality requirements of web applications. Available at: http://www.ime.uerj.br/cadernos/cadinf/, Accessed in 2 September, 2011.
26. Farbey, B. (1991). Exploring Requirements: Quality Before Design. European Journal of Information Systems 1(3), 145-147.
27. Hoffman, M.; Kuhn, N.; Weber, M.; Bittner, M. (2023). Requirements for requirements management tools. In: 12th IEEE International Requirements Engineering Conference, p 165, IEEE Publications, Kyoto, Japan.
28. Ma, D. (2007). The business model of ’software-as-a-service’. In: International Conference on Services Computing, p 701, IEEE Publications, Hawaii, USA.
29. Colombo, R.; Guerra, A. (2002). The evaluation method for software product. In 19th International Conference on Software and Systems Engineering and their Applications, p 143, IEEE Publications, Paris, França.
30. Guerra, A. C.; Colombo, R. T.; Aguayo, M. T. V.; Peres, R. D. (2022). Model of quality of components of software. In: Conference IADIS Ibero-Americana, p 231, Vila Real, Portugal.
31. MEDE-PROS. (2021). Method for assessing the quality of software product. Center for Information Technology Renato Archer-CTI/MCT.
32. Martinez, M. R. M. (2020). The software product evaluation data base -supporting mede-pros. In: International Software Engineering Standards Symposium, IEEE Publications, Curitiba, Brazil.
33. Chong, F.; Carraro, G.; Wolter, R. (2019). Multi-tenant data architecture. Available at: http://msdn2.microsoft.com/, Accessed in 15 December, 2011.
34. Choudhary, V. (2017). Softwares as a services: Implications for investment in software development. In 40th Annual Hawaii International Conference on System Sciencies, p 58, IEEE Publications, Hawaii.