Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 2, 2026

ANALYSIS OF SOFTWARE QUALITY METRICS FOR CLOUD COMPUTING SYSTEMS

Fatima Salahuddin[1], Komal Bashir[*2]

[1,*2]*Lahore College for Women University, Lahore*

[*2]komal.bashir@gmail.com

DOI: https://doi.org/10.5281/zenodo.18739381

Keywords
Analysis of Quality Metrics, Cloud Computing , Quality metrics , Software quality

Article History
Received: 24 December 2025
Accepted: 08 February 2026
Published: 23 February 2026

Corresponding Author: *
Komal Bashir

Abstract
In recent years, several suggestions have been made on how to quantify the quality of cloud services. However, there is still no clear guideline for stakeholders on which criteria to use. This work systematically maps cloud computing quality-of-service (QoS) and associated software quality measures to unify the field, provide a taxonomy, and establish practitioner-focused selection criteria. We started by looking through the literature to find possible variables and measurements. We then conducted a two-phase industry survey with senior cloud professionals: Survey-1 collected answers from 30 enterprises to see how useful and relevant the elements were for creating Criteria for using software quality metrics in cloud systems. A second study asked 26 businesses to rank these characteristics based on their requirements and judge how well an alternative metrics approach that uses the Criteria works. Lastly, we tested the Criteria with 23 firms that had previously put metrics in place for cloud-based software development. We got input using a questionnaire and a follow-up email to check for clarity, completeness, and perceived usefulness. The paper provides (i) a structured taxonomy of cloud service quality metrics, (ii) a practitioner-prioritized set of adoption criteria, and (iii) an examined alternative metrics model aimed at facilitating consistent, evidence-based selection of cloud service quality measures in practice.

1. Introduction
Software quality metrics are part of software metrics. It focuses in particular on the product quality, process, and viewpoint of the project. It is more associated with the product and process than the project metrics. Software quality metrics are divided into two parts. The first is end-product quality, and the second is in-process metrics. Metrics are used in critical systems, for example, in airline traffic control systems, weapons, and avionics. These metrics measure the defect relative to the system size and are expressed in the form of lines of code or function points. It is also used in many commercial software systems [1],[2].
The growing interest in addressing the problems associated with the quality of cloud services has led in recent years to the proposal of different metrics to assess the quality of cloud services. Metrics provide useful information for technical, operational, and market decisions that can be calculated and used in the enterprise [3].
A modern organizational and business model for the tech industry is cloud computing. Agility, lower entry costs, system independence, location independence, and scalability are the benefits of cloud computing over conventional computing. Many cloud service providers have arisen because of the benefits, and consumers are expanding their adoption [4].
Cloud computing services' quality is becoming a top priority concern, irrespective of the models of cloud delivery. Moreover, the acceptance of cloud services would depend on the ability to certify the quality of service. The current web

service quality metrics were updated and extended to cloud computing services [5].
Meanwhile, several quality attributes consist of the quality metrics. The attributes have different weights and communicate with each other for the overall output of the cloud computing service. Therefore, to correctly assess and direct the mechanism, we must consider the dynamic interrelationships between them [6].
This research proposes a mechanism for analyzing the Cloud Computing systems' performance. Additionally, it can also help to analyze the objective metrics that allow architects to measure quality characteristics such as scalability, elasticity, and performance.
Software Quality is an abstract concept. It can be hard to recognize its presence, but its absence can be immediately seen. Therefore, to increase software quality, the definition of software quality must first be addressed. It describes the quality of software in the context of software engineering and measures the software's performance (design quality) and how well the software complies with that design (quality of conformance). The "fitness for functioning" of a piece of software is also defined. [7].
The ISO/IEC 25010 standard gives a valuable model of 8 programming quality measurements. Dimensions of software quality are Maintainability, portability, functionality, performance, compatibility, usability, reliability, and security [8],[9].
Metrics are quantitative measures widely used for performance or output assessment, comparison, and monitoring [11]. Generally speaking, a group of metrics is used to build the dashboard that management or analysts periodically analyze to maintain performance reviews, perceptions, and business strategies [12], [13].
A variety of comprehensive indicators have been developed for ongoing assessments by best practices across the industry. However, the choice of measurements generally depends upon particular cases and scenarios [14].
Software metrics are an indication of some property or parameter of a software piece. The goal is to achieve the objective, reproducible, and observable calculation with a broad range of useful applications in the timetable and budget planning, cost estimates, quality assurance checks, software debugging, optimized software efficiency, and optimum tasks for staff [15].
Software metrics are central to developer processes; explicitly present variances between expected or successful performance; forecast circumstances requiring management intervention, early identification, and prediction; support an evaluation of the impact of proposed program changes [16].
Software measurements provide the project managers with reliable knowledge. In the entire software organization, Metrics offers quantitative knowledge [17]. The complexity that often accompanies complex and small software projects is reduced. Measurement allows managers to identify and prioritize project challenges, monitor them, and communicate them at all levels. The status of software project processes and products can be defined precisely with metrics. The progress of projects and the quality of associated software products during the project life cycle must be objectively portrayed [18].
Software quality metrics are a subset of software metrics focusing on product, process, and project quality aspects [19]. The processes and product metrics are more closely related than project metrics [20]. The most common metric of software quality is code line error, test cover, complexity, run time (performance), memory use, size of the software (binary file size), mutability aging, fault reply time (default report resolution time), step containment, or fault escapes (defects that were created in a particular phase but detected in a later phase) [21], [22].

Cloud Computing

It is a paradigm for easy, on-demand network access to a typical pool of configurable processing assets that can be easily provisioned and distributed with negligible management overhead (e.g., systems, software, servers, storage, and services) [23].
Cloud computing services' quality is becoming a top priority concern, irrespective of the models of cloud delivery; the acceptance of cloud services would depend on the ability to certify users' quality of service. The current web service quality metrics were updated and extended to cloud computing services [24].
Cloud computing helps to eliminate large sums of CAPEX (capital costs) & OPEX (Operating

expenses). A company does not need to spend on costly infrastructure, storage devices, applications, etc. It helps to cut and increase the demand for resources according to your needs. For example, if there is heavy web traffic, it will boost the services and vice versa [23]. Cloud computing gives the ability to have an Internet connection from wherever and whenever required. Updates and enhancements to software because both software maintenance and updates manage the cloud infrastructure through the cloud service provider [25],[26].

Role of Software quality metrics in cloud computing

The trust properties demonstrate the capabilities of cloud providers. The point of the structure was to check the properties displayed by the CAIQ (The Consensus Assessment Initiative Questionnaire aims to help organizations document and assess the security controls of cloud providers), and to give a response regarding the survey of the cases of cloud suppliers. The developers have familiarized themselves with the scientific classification with which they prepare and arrange CAIQ controls on trust assets, including sort control and approval specialists [26].

Metrics (performance, economics, security, and general) were categorized into four groups. For each category, a set of metrics was then proposed with which to quantify specific features. From the viewpoint of service providers, the assessment included the major service categories (SaaS, PaaS, and IaaS). The most significant drawback of this analysis metric (e.g., reliability, legibility, and service modularity) has been given, and how these metrics can be calculated is unclear. Other surveys aimed at evaluating and classifying cloud metrics have also been released [11], [12].

Researchers suggested performance appraisal for the taxonomy of commercial cloud services. In two dimensions, this taxonomy was constructed: quality function and experiment. In addition, the performance role was broken down into four components and seven elements of capability. The experimental factor was further broken down into five environmental scenes and fifteen organizational scenes (e.g., human sub-processes). Even though this scientific categorization helps assess current appraisal practices and grow new investigations, it offers, as it were, seven measurements, one for each size part, and is confined to execution assessment [10].

Metrics (performance, economics, security, and general) were categorized into four groups. For each category, a set of metrics was then proposed with which to quantify specific features. From the viewpoint of service providers, the assessment included the major service categories (SaaS, PaaS, and IaaS). The most significant drawback of this analysis metric (e.g., reliability, legibility, and service modularity) has been given, and how these metrics can be calculated is unclear. Other surveys aimed at evaluating and classifying cloud metrics have also been released [11].

The output was taken, and several 'Quality of Service' methods and parameters were implemented that could be used to measure this function. Nine properties and their related measurement approaches were focused on the Quality of Service criteria, while the procedure represented the methods, techniques, and processes (e.g., scheduling and detecting) available to confirm performance. Even though the study talked about some current execution estimation approaches, this is a casual study, and a more exhaustive survey is needed to investigate the current estimates used to evaluate both execution and different highlights of Quality of Service [27].

Taxonomy suggested for cloud metrics, the focus was on four system features: the isolation of performance between the clients and the resultant consistency in performance, and the elasticity of the cloud infrastructure. For evaluation and examination measurements, the scientific classification proposed four levels of reflection (for example, traditional productivity measurements, cloud innovation measurements, strategy measurements, and the executive's choice measurements). Its purpose was to allow cloud offerings and technologies to be compared and to provide providers, consumers, and end-users with a shared understanding [27].

Cloud services are important to elasticity, and a variety of studies concentrate on how to calculate it. The authors defined both the elastic resource management criteria and problems and potential solutions for the PaaS provider [28]. Coutinho proposed elasticity measurement concepts, metrics, and instruments. The elasticity of an isolated property was tested in both experiments. However, this is not adequate to take into account the effect this property has on other quality characteristics, such as scalability and reliability of the cloud service [29].

A systematic literature review was conducted that focused on a particular collection of quality and metrics. The authors received a 97-meter catalog focused on assessing the efficiency, economy, and protection of cloud service. This resulted in a system concept to enable the selection of IaaS and PaaS commercial cloud services, but not SaaS [30]. Later, Li et al. used PLR (previous literature review) as a baseline and expanded it to analyze the processes, properties, measurements, benchmarks, and test environments involved in the assessment of commercial cloud services. The findings showed that current works used several metrics, in addition to the cost of business cloud services, to calculate different performance characteristics [31].

A systematic analysis of literature was undertaken to explore cloud service scalability, elasticity, and performance concepts and metrics. They were only supplied by Google Scholar. The result showed that these attributes had a shared definition and suggested metrics [32].

Scheuner & Leitner presented a summary of the performance assessment of Feature as a Service (FaaS). FaaS offers a radically new cloud infrastructure model that enables serve less architecture to be achieved (micro-services). Dynamic resource provision and auto-scaling are some of their advantages [33]. This study was focused on academic and grey literature and looked at existing patterns, configurations of platforms, and performance features. The performance characteristics were therefore limited to four attributes (i.e., overload network, competitors for working load, length of instances, and inspection of infrastructure). No metrics were collected [34]. To define and classify existing QoS information in fog computing, a systematic literature review was conducted. Fog Computing decentralizes services and resources both outside and close to the cloud. Three categories (communication, application, and service) with eleven QoS factors were covered [35]. This work (i.e., energy consumption, throughput, resource utilization, deadline, cost, response time, reliability, execution time, scalability). The authors rated the value of the use of the QoS factor. The most widely used response time, costs, and usage of resources are availability and scalability. Again, the study does not completely cover characteristics of consistency, and measurements have not been obtained [11].

2. Materials and Methods

This goal was to systematically define and taxonomically distinguish available data on quality indicators for cloud services, as well as to provide such a holistic analysis to assess the current research's possible limitations.

This study aims to add to the body of knowledge about cloud service evaluation by providing a set of quality metrics for cloud services; identifying the quality attributes that measure the selected metrics, aligning them with the quality characteristics and cloud computing concepts such as service models, and identifying limitations in evaluating cloud services.

Requirements for scalability, elasticity, and efficiency come up frequently when software architects design cloud applications and when cloud consumers and providers negotiate service-level objectives. This research targets these three properties as their definition is unclear. For measuring the scalability, elasticity, and efficiency of cloud environments, metrics are necessary in addition to definitions of the three properties.

Therefore, the research questions RQ1 targeting property definitions and RQ2 targeting metrics for the three properties.

RQ1: Which are the factors that are closely related to software quality metrics in the cloud computing context, and to what extent are these factors fulfilled by this?

RQ2: “How to define scalability, elasticity, and efficiency in the context of cloud computing?”

RQ3: “What metrics can be used to compare the scalability, elasticity, and efficiency of different cloud computing services, and how are they measured and used?”

RQ4: To what extent are these factors fulfilled by the alternate model/s?

RQ5: "What is the priority of the Organizations?"

The primary objective of this work was to devise Criteria for the adoption. Literature review with the detailed survey based on Google Forms is being selected for finding the answers to Research Question 1. The literature review is used to find the factors that are needed for the successful adoption of this method for the software development process.

The industrial survey for collecting the evidence information about the effectiveness of the factors listed below is conducted by the organizations that are already working on these metrics.

The proposed research method process is presented in Figure 1.

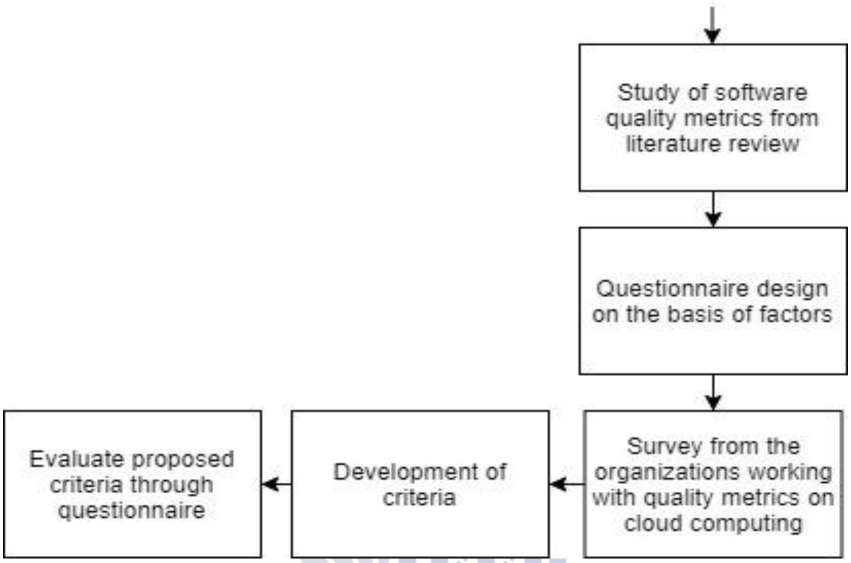

Figure 1: The Process Steps of the Research Method

Listed below are the factors extracted from the literature review:

- Quality of the code
- Lead time of projects
- Availability of the system under development
- Reduction of workforce size
- The capability of employees to solve the problem
- Scalability of the system under development
- Elasticity of the system under development
- Efficiency of the system under development
- Customer feedback

By using these factors, a survey was conducted among the organizations that are already practicing their software development process, which ensures the effectiveness of these factors by giving a score against each factor based on their experience.

Quality factors that are extracted from the literature review are explained in Figure 2.

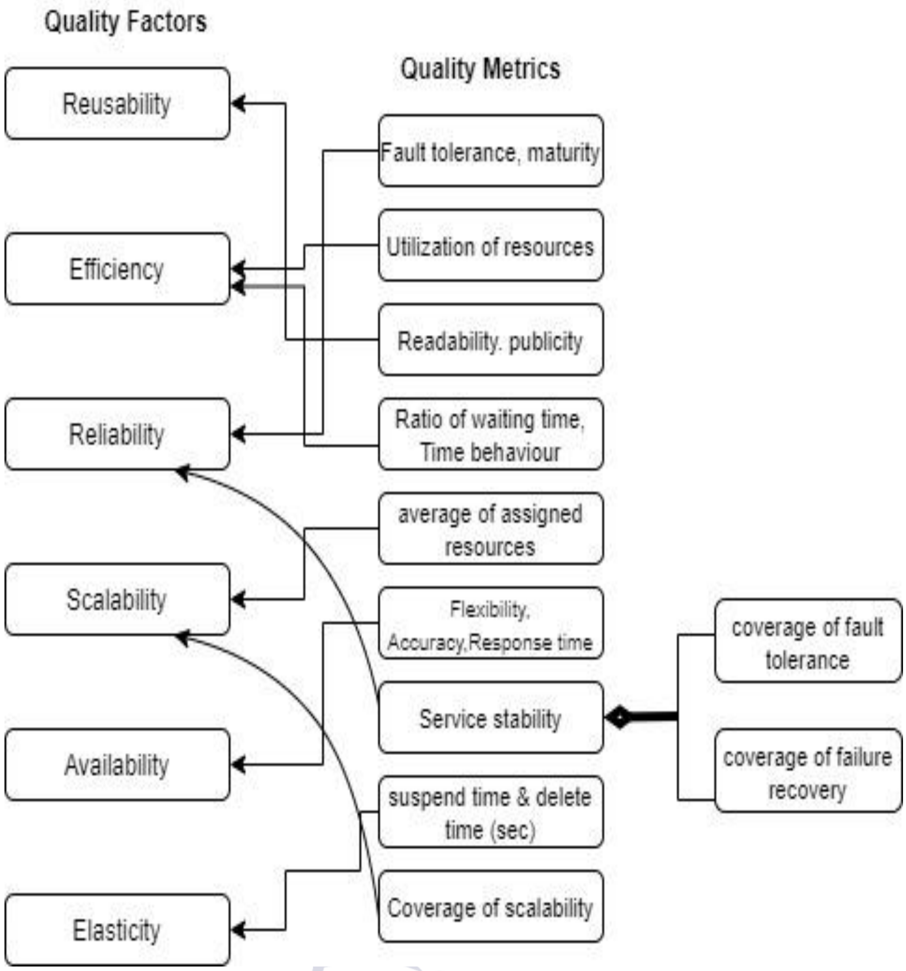

Figure 2: Quality factors for Software Quality Metrics

3. Results and Discussion

The survey results are analyzed by graphs and tabular form, which are categorized based on percentages. A scale of 1-5 is used for the rating of factors, with 1 being low and 5 being high. The survey was sent to 55 different organizations.

Forty-three (43) respondents gave their responses, out of which 42 are those who already practice quality metrics for scalability, elasticity, and efficiency for the software development process. And the results of those 42 questionnaires are considered for the Criteria making.

The following are the questions asked in the survey, and Table 1 shows the results obtained against these questions.

Questions

1. To what extent has DevOps improved the quality of the Code in your projects? (In terms of maintainability and re-usability of code)
2. To what extent has Quality Metrics reduced the lead time of cloud projects?
3. To what extent has DevOps increased the availability of the system under development?
4. To what extent has the use of software quality metrics reduced the workforce size (i.e., number of members in the team)?
5. To what extent have software quality metrics enhanced the capability of employees to solve the problem through knowledge sharing?
6. The Improvement in the Elasticity of the system under development?
7. To what extent has Efficiency improved in the system under development?
8. To what extent have quality metrics improved the scalability of the system under development?
9. To what extent have metrics helped in getting rapid feedback from the customers as well as the internal team?

Table 1: Results of Survey I

Q.No.	Very Low	Low	Moderate	High	Very High
1.	3	5	6	17	11
	7.12%	11.90%	14.28%	40.50%	26.20%
2.	2	5	10	21	4
	4.80%	11.95%	23.80%	50%	9.50%
3.	1	2	10	16	13
	2.40%	4.80%	23.80%	38.10%	30.90%
4.	1	7	13	18	3
	2.38%	16.67%	30.95%	42.85%	7.12%
5.	5	5	6	13	13
	11.90%	11.90%	14.28%	30.96%	30.96%
6.	1	6	12	6	17
	2.38%	14.28%	28.56%	14.28%	40.50%
7.	2	3	6	20	11
	4.76%	7.17%	14.28%	47.62%	26.20%
8.	1	5	4	17	15
	2.38%	11.90%	9.51%	40.50%	35.71%
9.	1	4	6	13	18
	2.38%	9.51%	14.28%	30.98%	42.85%

In survey-1 results of only those organizations are being considered that have already been using quality metrics for their software development process. Based on the experience of those organizations, the effectiveness of metrics in improving cloud computing projects, the following factors are being concluded.

The average rating of the factors is being considered. Table 2 depicts the effectiveness, where

Column 1 shows the factors that are being extracted from the literature review.

Column 2 shows the labeling of the factors.

Column 3 is the effectiveness of quality metrics according to these factors.

Table 2: Results of Survey II

Factors	Label	Effectiveness
Quality of the code	F1	4
Lead time of projects	F2	4
Availability of the system under development	F3	4
Reduction of workforce size	F4	4
The capability of employees to solve the problem	F5	5
Elasticity of the system under development	F6	5
Efficiency of the system under development	F7	4
Scalability of the system under development	F8	4
Customer feedback	F9	5

Survey II was sent to 35 organizations. 26 organizations recorded their responses, out of which 23 are those who already are implementing quality metrics in cloud computing for the software development process, and 3 organizations are those who have not yet adopted the model. The purpose of this survey is to get the priority list of organizations, which factor is important to them, which are less important, and secondly, to get the effectiveness of their previous/current model that is being used as the alternative.

As a result of this questionnaire, we get the priority list and effectiveness of the previous model, which are amended with the values acquired from survey 1.

The priority and effectiveness of the alternate model for each organization are different, so the results of each organization are different from those of the other.

For validation criteria were sent to the 26 organizations (through Email and questionnaire), and 20 organizations responded to it. Out of 20, eighteen are those respondents who agreed with the results of our Criteria. According to our Criteria, Organization-5 should continue using the current model, but they decided to adopt quality metrics methodology, and Organization-25 should adopt quality metrics methodology. However, when it was being inquired by the organization, they said that they are getting satisfactory results from their current model and they don't need to adopt quality metrics methodology. By comparing the results, it is concluded that 18 out of 20 respondents agreed with the results of our criteria, which depict that this work has an accuracy of **90%.**

4. Conclusion

This research thoroughly categorized and evaluated external and internal quality indicators for cloud services for software development organizations, using a structured literature review. The results indicate a taxonomy that organizes the present array of metrics. We created a single reference that makes it clear what to measure, how to assess it, and where there are still gaps by mapping current measurements to this taxonomy and looking at their breadth, granularity, and evidence of usage.

The metric suggestions that resulted from this research are useful for software architects and engineering teams working on cloud applications. They help make evaluations more uniform and help build new measurements that are suited for the situation. The taxonomy creates a common language for both providers and consumers that may help with service objectives (such as SLAs/SLOs), make discussions less confusing, and make it easier to compare different services.

With an increase in cloud providers and products, each with its own pricing, features, and guarantees, it becomes harder to find a provider that meets strict QoS requirements. Especially when it comes to privacy and performance. Our framework helps organize that decision-making process by connecting corporate goals with quantifiable standards and pointing out trade-offs that may not be obvious.

There are several problems with this work: it is based on published evidence that was accessible at the time of the evaluation and may not show new, proprietary, or domain-specific techniques. Subsequent research must authenticate and enhance the taxonomy within longitudinal industry contexts, integrate dynamic and real-time observability signals, expand the scope of privacy, security, and compliance metrics, and establish tool support (including checklists, dashboards, and automated evidence collection) to facilitate the operationalization of metric selection and monitoring at scale.

5. References

G. K. Thiruvathukal, N. J. Hayward, and K. Läufer, "Metrics dashboard: A hosted platform for software quality metrics," arXiv:1804.02053, 2018.

P. Kokol, "Software quality: How much does it matter?," Electronics, vol. 11, no. 16, p. 2485, 2022.

M. Becker, S. Lehrig, and S. Becker, "Systematically deriving quality metrics for cloud computing systems," in Proc. 6th ACM/SPEC Int. Conf. Performance Engineering (ICPE), 2015, pp. 169–174, doi: 10.1145/2668930.2688043.

S. Dalla Palma, D. Di Nucci, F. Palomba, and D. A. Tamburri, "Toward a catalog of software quality metrics for infrastructure code," Journal of Systems and Software, vol. 170, p. 110726, 2020, doi: 10.1016/j.jss.2020.110726.

N. B. Kassie and J. Singh, "A study on software quality factors and metrics to enhance software quality assurance," International Journal of Productivity and Quality Management, vol. 29, no. 1, pp. 24–44, 2020.

C. R. Choi and H. Y. Jeong, "Quality evaluation and best service choice for cloud computing based on user preference and weights of attributes using the analytic network process," Electronic Commerce Research, vol. 14, no. 3, pp. 245–270, Dec. 2014, doi: 10.1007/s10660-014-9156-1.

J. P. Cavano and J. A. McCall, "A framework for the measurement of software quality," in Proc. Software Quality Assurance Workshop on Functional and Performance Issues, 1978, pp. 133–139.

M. H. Samadzadeh and K. Nandakumar, "A study of software metrics," Journal of Systems and Software, vol. 16, no. 3, pp. 229–234, 2016, doi: 10.1016/0164-1212(91)90017-Z.

A. Jeanrenaud and P. Romanazzi, "Software product evaluation metrics: A methodological approach," WIT Transactions on Information and Communication Technologies, vol. 9, 2025.

Z. Li, L. O'Brien, R. Cai, and H. Zhang, "Towards a taxonomy of performance evaluation of commercial cloud services," in Proc. 2012 IEEE 5th Int. Conf. Cloud Computing (CLOUD), 2012, pp. 344–351.

A. K. Bardsiri and S. M. Hashemi, "QoS metrics for cloud computing services evaluation," International Journal of Intelligent Systems and Applications, vol. 6, no. 12, pp. 27–33, 2014.

R. P. L. Buse and W. R. Weimer, "A metric for software readability," in Proc. 2008 Int. Symp. Software Testing and Analysis (ISSTA '08), 2018, pp. 121–130, doi: 10.1145/1390630.1390647.

L. Langfeldt, I. Reymert, and D. W. Aksnes, "The role of metrics in peer assessments," Research Evaluation, vol. 30, no. 1, pp. 112–126, 2021.

S. Chulani et al., "Metrics for managing customer view of software quality," in 5th Int. Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No. 03EX717), 2004, pp. 189–198.

J. Jiarpakdee and C. Tantithamthavorn, "The impact of correlated metrics on the interpretation of defect models," IEEE Transactions on Software Engineering, pp. 1–1, Jan. 2019, doi: 10.1109/TSE.2019.2891758.

K. P. Srinivasan and T. Devi, "Software metrics validation methodologies in software engineering," International Journal of Software Engineering & Applications, vol. 5, no. 6, p. 87, 2014.

H. F. Li and W. K. Cheung, "An empirical study of software metrics," 2015. (Venue/details not provided.)

N. E. Fenton and M. Neil, "Software metrics: Successes, failures and new directions," Journal of Systems and Software, vol. 47, nos. 2–3, pp. 149–157, 1999.

A. B. Al-Badareen, M. H. Selamat, M. A. Jabar, and J. Din, "The impact of software quality on maintenance process," International Journal of Computer, vol. 5, no. 2, pp. 1–8, 2011. [Online]. Available: https://www.researchgate.net/publication/237135177

D. Galin, "Software product quality metrics," IEEE Transactions on Software Engineering, vol. 4, pp. 25–28, 2018.

G. L. Saini and D. Panwar, "Comparative study of open source software quality models," Journal of Intelligent Systems and Computing, vol. 1, no. 1, pp. 46–55, 2020.

H. Tu, W. Sun, and Y. Zhang, "The research on software metrics and software complexity metrics," in Proc. 2009 International Forum on Computer Science-Technology and Applications (IFCSTA 2009), 2019, vol. 1, pp. 131–136, doi: 10.1109/IFCSTA.2009.39.

S. Whiteside and J. Lucas, "Cloud computing: The next revolution in process safety," Hydrocarbon Processing, 2015-June (May issue), 2015.

N. Upadhyay, "Managing cloud service evaluation and selection," Procedia Computer Science, vol. 122, pp. 1061–1068, 2017, doi: 10.1016/j.procs.2017.11.474.

J. Weinman, "The future of cloud computing," in IEEE Technology Time Machine Symposium on Technologies Beyond 2020 (TTM 2011), 2011, doi: 10.1109/TTM.2011.6005157.

J. B. Horrigan, "Cloud computing applications and services," Explorations: The Journal of Science and Healing, vol. 4, no. 3, pp. 165–167, 2008.

N. Herbst et al., "Quantifying cloud performance and dependability: Taxonomy, metric design, and emerging challenges," ACM Transactions on Modeling and Performance Evaluation of Computing Systems, vol. 3, no. 4, 2018, doi: 10.1145/3236332.

F. D. Muñoz-Escoí and J. M. Bernabéu-Aubán, "A survey on elasticity management in PaaS systems," Computing, vol. 99, no. 7, pp. 617–656, 2017.

E. F. Coutinho, F. R. de C. Sousa, P. A. L. Rego, D. G. Gomes, and J. N. de Souza, "Elasticity in cloud computing: A survey," Annals of Telecommunications, vol. 70, no. 7, pp. 289–309, 2015.

Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a catalogue of metrics for evaluating commercial cloud services," in Proc. IEEE/ACM Int. Workshop on Grid Computing, 2012, pp. 164–173, doi: 10.1109/Grid.2012.15.

Z. Li, H. Zhang, L. O'Brien, R. Cai, and S. Flint, "On evaluating commercial cloud services: A systematic review," Journal of Systems and Software, vol. 86, no. 9, pp. 2371–2393, 2013, doi: 10.1016/j.jss.2013.04.021.

S. Lehrig, H. Eikerling, and S. Becker, "Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics," in Proc. 11th Int. ACM SIGSOFT Conf. Quality of Software Architectures (QoSA 2015), part of CompArch 2015, 2015, pp. 83–92, doi: 10.1145/2737182.2737185.

J. Scheuner and P. Leitner, "Function-as-a-service performance evaluation: A multivocal literature review," Journal of Systems and Software, vol. 170, 2020, doi: 10.1016/j.jss.2020.110708.

A. de Vaulx, "Cloud computing service in metrics description," NIST Special Publication 500-307, no. 307, p. 24, 2018. [Online]. Available: https://doi.org/10.6028/NIST.SP.500-307

M. H. Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," International Journal of Communication Systems, vol. 33, no. 8, pp. 1–34, 2020, doi: 10.1002/dac.4340.