Big Data Fortaleza Platform: Quality Improvement with Testing Process

Amanda K. B. Cavalcante[1], Ícaro S. de Oliveira[1,2], Victória T. Oliveria[1], Pedro Almir M. Oliveira[1,3], Tales P. Nogueira[1,4], Ismayle S. Santos[1,2] and Rossana M. C. Andrade[1]

[1]*Computer Networks, Software and Systems Engineering Group (GREat), Federal University of Ceará (UFC), Fortaleza, CE, Brazil*

[2]*State University of Ceará (UECE), Fortaleza, CE, Brazil*

[3]*Federal Institute of Education, Science and Technology of Maranhão (IFMA), São Luís, MA, Brazil*

[4]*University of International Integration of Afro-Brazilian Lusophony (Unilab), Redenção, CE, Brazil*

{*amanda.cavalcante, icarooliveira, victoria.oliveira pedromartins* }*@great.ufc.br, tales@unilab.edu.br,*

Keywords: Quality, Testing, Platform, Big Data.

Abstract: In July 2022, the City Planning Institute of Fortaleza (Iplanfor), in collaboration with Computer Networks, Software and Systems Engineering Group (GREat) from the Federal University of Ceará, launched a project to develop a platform utilizing Big Data for data analysis and predictive modeling. This initiative aimed to support strategic planning and create solutions that would foster the development of City Fortaleza, ultimately guiding public policies based on solid evidence. The platform was named Big Data Fortaleza. Given its focus on government applications, it was essential to validate the platform through various testing methods. This article outlines the adopted testing process and highlights critical outcomes, including improved prediction accuracy and enhanced system and data security efficiency. Additionally, it discusses valuable lessons learned, such as the importance of effective team communication and the necessity for ongoing adjustments to maintain the platform's quality and reliability.

1 INTRODUCTION

In recent years, integrating advanced technologies has played a crucial role in developing and enhancing intelligent cities. According to Washburn et al. (2010), an intelligent city should utilize computing technologies to make essential components and services of city infrastructure — such as municipal administration, education, healthcare, public safety, real estate, transportation, and utilities—smarter, more interconnected, and more efficient. In this context, in July 2022, a collaboration between the City Planning Institute of Fortaleza and the Federal University of Ceará led to creating a project to bring Fortaleza closer to becoming a smart city: the Big Data Fortaleza Platform. This project emerged from the need to leverage Big Data technologies to analyze complex data and generate valuable insights that can assist public administration in strategic planning and urban development for the city.

The Big Data Fortaleza platform, as previously described and studied in (Santos et al., 2023), (Costa et al., 2024), and (Élcio Batista et al., 2024), was designed not only to collect and store large-scale data but also to utilize advanced analysis and prediction techniques. Its primary objective is to provide essential information for decision-making processes, particularly in the formulation and implementation of public policies based on solid evidence. The development of this platform was accompanied by a rigorous testing process to ensure its functionalities are efficient and reliable, as well as to maintain the security and integrity of the data it manages.

This article will present an overview of the testing process adopted, the challenges of testing Big Data systems, and the results obtained. Additionally, the importance of incorporating testing from the early stages of development will be discussed, highlighting the lessons learned throughout this process.

This report provides insight into the testing process of the Big Data Fortaleza platform. It serves as a guide for development teams facing similar challenges in both the public and private sectors, emphasizing the importance of testing in Big Data systems to deliver more reliable and high-quality outcomes.

This article is structured into six main sections.

Cavalcante, A. K. B., S. de Oliveira, Í., Oliveria, V. T., Oliveira, P. A. M., Nogueira, T. P., Santos, I. S. and Andrade, R. M. C.
Big Data Fortaleza Platform: Quality Improvement with Testing Process.
DOI: 10.5220/0013290500003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 2*, pages 227-236
ISBN: 978-989-758-749-8; ISSN: 2184-4992

In Section 2, related works are discussed, providing an overview of existing research and practices in the field. Section 3 contextualizes the project, describing its environment and motivations. Section 4 details the actions taken, including the methods and procedures adopted. Section 5 presents the results achieved with the project implementation. In Section 6, we report some insights from a survey conducted with professionals involved in the system's development. Finally, Section 7 discusses the lessons learned, highlighting insights and recommendations for future work in the area.

2 RELATED WORKS

Quality assurance (QA) research for big data systems has grown significantly recently. Various studies have explored different aspects and approaches to ensure data quality and the effectiveness of extensive data systems. This section reviews the most relevant works contributing to understanding and advancing QA practices in big data. To find those works, a literature review was conducted using the *Scopus* database, known for its extensive coverage of software engineering research. The selection of related studies involved a search string with synonyms, including "big data systems," "big data platform," "experience report," "case study," "implementation report," "field report," "lessons learned," "practical experience," "quality assurance," QA, and "software testing." These terms aimed to filter studies on testing and quality in these systems. The decision to use *Scopus* was based on its coverage of significant digital libraries and the use of IEEE Xplore, which aligns with the recommendations of other systematic reviews in software engineering (Staegemann et al., 2020).

The studies by (Daase et al., 2024) and (de Oliveira et al., 2024) conducted systematic reviews to address research gaps by identifying and cataloging specific methods, techniques, practices, and tools for quality assurance in big data systems. (Daase et al., 2024) particularly emphasized the challenges of testing in big data environments, including issues related to realistic datasets and scalability. Building on their work, our review systematically categorizes existing QA methods and tools, assesses their effectiveness, and identifies best practices tailored to the unique demands of big data systems. These approaches have been instrumental in the development of this report.

Another relevant study in the field of quality assurance for systems handling data is by Nasir, Neelum, et al. (Nasir et al., 2022), titled "Testing Framework for Big Data: A Case Study of Telecom Sector of Pakistan." This work proposes a specific testing framework for big data that can be applied to the telecommunications sector in Pakistan. The authors develop and implement a set of testing practices and tools tailored to big data's unique requirements and challenges, such as volume, variety, and velocity of data. The case study demonstrates the framework's effectiveness in improving data quality and enabling early fault detection, contributing to the stability and reliability of big data systems in the telecommunications sector.

In addition, we have the study by Punn, Narinder Singh, et al. (Punn et al., 2019), titled "Testing Big Data Application". This work explores testing strategies for big data applications, addressing the inherent challenges in verifying and validating systems handling large volumes of data. The authors propose a comprehensive testing approach that includes functional, performance, and security testing techniques specifically adapted to the unique characteristics of big data applications. The research emphasizes the importance of a robust testing infrastructure and the integration of automated tools to ensure the efficiency and effectiveness of the QA process.

Finally, we have the work "Deployment of a Machine Learning System for Predicting Lawsuits Against Power Companies: Lessons Learned from an Agile Testing Experience for Improving Software Quality" (Rivero et al., 2020). This study reports implementing a machine learning system to predict lawsuits against power companies, highlighting the lessons learned during an agile testing experience to improve software quality. The authors discuss the challenges faced, such as the need to adapt agile practices to accommodate the complexity of machine learning and the importance of continuous and iterative testing to ensure the system's accuracy and reliability. The lessons learned emphasize the importance of collaboration among multidisciplinary teams and flexibility in QA approaches to address the specificities of machine learning systems.

The review of related works demonstrates the diversity of approaches and methodologies applied to quality assurance in big data systems. Studies such as those by Nasir et al. (2022) and Punn et al. (2019) provide practical frameworks and strategies adaptable to different sectors and contexts. The experience reported in implementing machine learning systems for power companies underscores the importance of agile and collaborative practices. Together, these works offer a solid foundation and diverse perspectives that inform and enrich our QA approach in big data, especially in governmental environments, where the robustness of tools and the effectiveness of strategies are critical.

3 CONTEXT

The City Hall of Fortaleza developed a participatory strategic plan to integrate physical-territorial development with social and economic aspects. The plan seeks to bring together different perspectives and sectors and the various territories and levels of government in discussions about the city. The current challenges the plan aims to address are educational delays, low qualification levels, youth vulnerability, poverty, and social inequality (Santos et al., 2023).

One of the highlighted challenges is the issue of Early Childhood, which encompasses actions targeted at children from zero to six years old. This area of focus is cross-cutting, requiring interventions in health, education, social assistance, and other sectors. To address the challenges faced by the city of Fortaleza and its administration, data collection and analysis are essential for diagnosing problems and supporting municipal leaders' decisions. Through compelling analysis of this data, it is possible to improve service delivery to the community and predict the impact of changes in urban infrastructure. Thus, a robust infrastructure is essential for collecting, storing, and processing data from various sources [Ommited for Blind].

The creation of the platform *BigData X* enabled comprehensive analyses using data from various sectors related to early childhood. It delivered to municipal managers more than 20 *analytics* and three notification alerts linked to the departments of early childhood, education, health, social assistance, and drug prevention. As a result, municipal managers gained valuable *insights* for developing new public policies for citizens, such as integrating child vaccination programs into daycare centers. Within a month of the platform's launch, more than 2,000 children had their vaccination schedules updated and brought up to date.

In education, it was essential for public managers to ensure access to nurseries, daycare centers, and schools specialized in early childhood education. Regarding health, it was crucial to offer comprehensive care, encompassing prenatal and postnatal care for women and the initiation of newborn follow-up, including vaccine administration. In human rights and social development, it was necessary to identify and support families in socioeconomic vulnerability or homelessness, providing social benefits to reduce disparities and facilitate access to services and resources that promote the rights of children and pregnant women.

The development of the platform adopted the framework *Scrum*. *Scrum* is a widely used agile methodology in software development, characterized by its iterative and incremental approach through *Sprints* (Schwaber and Sutherland, 2020). The *Sprints* are well-defined, *timeboxed* periods lasting two to four weeks, with a clear objective and a set of product *backlog* items to be delivered by the end of the period.

In the case of *Big Data Fortaleza*, *Sprints* lasting two weeks were adopted after a discussion about the team's characteristics, such as the presence of *part-time* fellows. During this period, the teams worked in parallel and collaboratively, including requirements elicitation, screen design, data collection and extraction, platform development, and testing.

Testing, in turn, plays a crucial role in the context of *Big Data Fortaleza*, mainly due to the sensitive nature of the data involved. Given the vast amount of sensitive information collected and processed by the platform, quality assurance through testing is essential to mitigate security, privacy, and data integrity risks.

The tests encompassed three levels (Bourque and Fairley, 2014):

1. **Unit Testing:** focused on the isolated evaluation of software units and conducted by the development team;
2. **Integration Testing:** verified the integration between the various software units by the development team;
3. **System Testing:** analyzed the system as a whole to ensure its compliance with the established specifications and requirements.

In the following sections, the system testing process is described, including the tools used and the results and lessons learned during the platform's development period.

4 ACTIONS TAKEN

This section will detail the actions implemented to ensure quality in system development. Given the system's complexity and scale, it was essential to adopt a systematic approach that covered everything from requirements definition to final validation. We will describe unit, integration, and system testing, including functional and security tests. Additionally, we will present the tools used and the risk mitigation strategies. Each action described was essential to ensure the developed system's robustness, reliability, and effectiveness.

4.1 Unit and Integration Testing

For the system development, a *backend* design was chosen with the following technologies: Spring Boot[1], Spring Data[2], JPA[3], Postgres[4], Spring Security[5]. These technologies were essential due to the need to establish a robust access control policy for the platform.

Unit tests were crucial in assisting with the backend implementation, particularly for the creation of microservices, which are structured programs with well-defined inputs and outputs that are highly complex—for example, performing checks on texts and phrases with logical rules for numerical values.

The development team prioritized the implementation of integration tests in the system's backend. As government data is of utmost importance, the implemented business logic must be consistent, as it is a fundamental component in the processing and consumption of the data.

Docker with Test Container Postgres was used as a tool for integration testing with the database, creating something ephemeral specifically for that test. Additionally, JUnit5 was used to perform both unit and integration tests.

These tool and approach choices proved to be advantageous in the testing process. Docker with Test Container ensured the consistency and reliability of the tests. At the same time, the use of JUnit 5 and AutoConfigureMockMvc simplified the development and execution of the tests, allowing for precise validation of the business logic of the Big Data system.

API tests were also performed using the Postman tool[6], where requests were made to the system's API, and the response body, as well as the status code and its compliance with the documentation, were verified. Initially, due to the importance of data protection, the tests on the authentication controller and user management were prioritized. Moreover, the tool provides a history of executions, facilitating error monitoring.

4.2 System Testing

In this subsection, we address the system testing process, which is crucial for ensuring the integration and proper functioning of the platform's various features. System testing verified the entire system, ensuring that all components operate together as expected and meet the established requirements.

4.2.1 Functional and Interface Testing

In the process of functional testing for the Big Data system, test cases were initially created in Testlink[7], an open-source software test management tool, based on the platform's use cases. These were then manually executed following a plan also created within the tool. When a bug was detected, it was reported in GitLab[8], including a description, reproduction steps, severity, and evidence. After the bug was fixed, a retest was conducted to verify the effectiveness of the fix. This testing cycle can be summarized in Figure 1.

During functional testing, the system's compliance with the prototype created in Figma[9] by the design team was also verified. Additionally, exploratory testing was conducted, where testers used their experience and knowledge to explore different system usage scenarios. This was important for identifying potential flaws or unexpected behaviors that were not covered by traditional test cases. The combined approach of functional testing based on use cases and exploratory testing allowed for comprehensive coverage of the critical aspects of the system, contributing to its robustness and reliability in production environments.

4.2.2 Security Testing

Security testing aims to identify vulnerabilities in applications and can be divided into two categories (Aydos et al., 2022): (i) functional security testing, ensuring that the software's security functions are correctly implemented according to security requirements; and (ii) security vulnerability testing, focusing on discovering security vulnerabilities from an attacker's perspective. Thus, security testing involves an active application analysis to identify any weaknesses, technical flaws, or vulnerabilities (OWASP, 2023). In the Big Data Fortaleza system, tests were conducted to identify potential vulnerabilities in the platform due to the sensitivity of the information stored. In this regard, one of the security assessments followed these steps:

- **Step 1** Port and service scan on the platform host. Using the nmap[10] tool, the active ports on the

[1] https://spring.io/projects/spring-boot
[2] https://spring.io/projects/spring-data
[3] https://spring.io/projects/spring-data-jpa
[4] https://www.postgresql.org/
[5] https://spring.io/projects/spring-security
[6] https://www.postman.com/
[7] https://testlink.org/
[8] https://about.gitlab.com/
[9] https://www.figma.com/
[10] https://nmap.org/

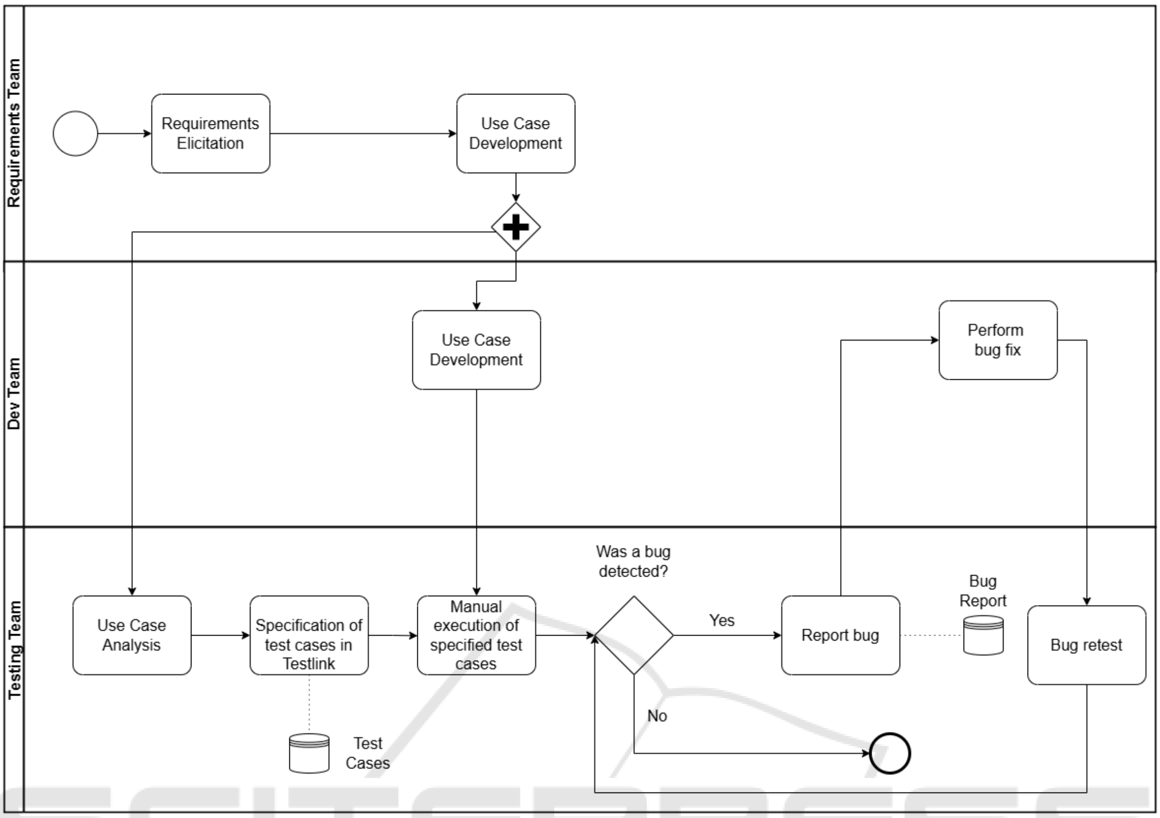

Figure 1: Functional testing process.

server where the platform was running were analyzed. After the scan, the versions of the running services were detected, and the known vulnerabilities associated with each version were analyzed using reports and vulnerability documentation platforms, such as NetApp[11] and CVEDetails[12].

- **Step 2** Web vulnerability scanner. The OWASP ZAP[13] tool was used to conduct a static analysis for vulnerabilities in the platform's code. Additionally, Burp Suite[14] was used to inspect the data and requests exchanged, focusing on identifying vulnerabilities in HTTP requests.
- **Step 3**. Security inspection. An inspection was carried out on the encryption algorithms used to check for deprecated algorithms or misconfigurations. Furthermore, other system libraries were inspected for security flaws.

[11] https://security.netapp.com/advisory/
[12] https://www.cvedetails.com/documentation/
[13] https://www.zaproxy.org/
[14] https://portswigger.net/burp

5 RESULTS

This section presents the results obtained during the application testing process. The goal is to provide a clear overview of the observed performance, critical issues identified, and the applied corrections. The analysis of these results highlights the final quality of the application and offers valuable lessons for future development and adjustments.

5.1 Results of Unit and Integration Tests

The unit and integration tests conducted on the application's *backend* proved effective in validating components and identifying issues early. The most relevant outcomes ara shown on Table 1.

During the development process, *Test-Driven Development* (TDD) was employed to write the tests. Initially, unit tests were created to verify the functionality of the libraries, followed by integration tests to ensure a more robust implementation. Tests were also separated at the controller level to focus specifically

Table 1: Test coverage in the backend.

Class	Method	Line
67% (474/700)	50% (2022/4038)	46% (5886/12598)

on routes and request responses more objectively. The implementation and coverage of these tests enhanced the security and quality of the development flow, ensuring that new implementations could rely on the stability of previous ones. So far, approximately 67% of the total lines of code have been covered by unit and integration tests. Unit tests conducted on the microservices achieved 100% coverage, as the code was written for specific tasks with fewer lines, allowing the tests to cover all existing functions efficiently.

5.2 System Testing Results

The results of the functional tests provide a detailed overview of the system's quality, with 228 test cases documented in Testlink. During the testing process, 194 bugs were identified, 166 of which have already been resolved by the system developers. A summary of these figures is presented in Table 2 .

Most of these bugs were related to minor interface issues, such as button names, section titles, and similar elements. However, critical and severe errors were also identified in key system functionalities. For instance, due to an issue with access management implemented in a late development stage, some user profiles lost access to functionalities they were supposed to have, such as viewing analytics.

One of the difficulties encountered was related to the instability of the bandwidth in the testing environment. It was observed that, due to the variability in the amount of data required to display analytics (graph, map, etc.), the connection to the Amazon Simple Storage Service (S3) was interrupted due to a *timeout*. Previously, the system handled each request for a route that processed multiple calls to S3. This caused a queuing of requests and delayed the response to generate the analytics, leading to errors and slow performance. Subsequently, after the implementation of microservices to specifically handle the search and loading of data, leaving the central system less overloaded, a reduction in the response time of the data and an improvement in the platform's performance were observed.

In the Big Data Fortaleza project context, where information security and data analysis play critical roles in public administration, security testing revealed significant vulnerabilities that were addressed before the system went into production. One notable example was the absence of a configuration for the Content Security Policy (CSP) header, used in HTTP requests, identified by the vulnerability scanner. In this case, the development team adjusted the configuration, ensuring the mitigation of attacks (including Cross-Site Scripting).

During this security evaluation, other aspects related to the management of cloud services and the protection of the Big Data system infrastructure were also observed, such as the need for services to handle Denial-of-Service (Anti-DDoS) attacks and the importance of creating rules to prevent data leakage via unauthorized access to team accounts. As a result, several actions were taken:

- Strengthening the critical rotation policy and using Multi-Factor Authentication (MFA).
- Restricting incoming and outgoing traffic to ensure that the City Hall's firewall would initially filter all accesses.
- Encrypting data stored in the cloud.
- Performing daily backups with a 30-day retention period.
- Strengthen firewall rules to prevent bot access and the exploitation of known vulnerabilities.
- Adopting AWS Shield for DDoS protection.
- Using CloudTrail to log all actions performed on the platform's supporting infrastructure.
- Utilizing AWS GuardDuty to analyze potential threats.

6 DATA QUALITY

Data is essential to modern life (Wang et al., 2023) and is considered an asset that aids in strategic business and policy decision-making based on data *insights* (Taleb et al., 2016). This relevance has contributed to the emergence of data-driven decision-making, which prescribes that data is at the core of decision-making and influences the quality of decisions (Wang et al., 2023).

Wang et. al (2023) conducted a literature review to understand the dimensions of data in such a way that they can be of high quality, as they are essential for decision-making. The authors summarized the findings into 21 data quality dimensions, with five being the most important: completeness, accuracy, timeliness, consistency, and relevance. In this context, it is vital to understand the significance of data and how

Table 2: Quality Monitoring: Number of Tests and Fixes.

Number of test cases	Number of reported bugs	Number of fixed bugs	Number of automated API tests
228	194	166	69

it can contribute to decision-making aimed at improving software quality, impacting both the quality of the product and the process.

Given this, a survey was conducted internally with the development professionals, data scientists, and the quality team of the project to understand their perceptions regarding the main challenges in pursuing data quality, considering the five attributes mentioned above.

Based on the responses from the form, it was possible to collect 11 responses with various insights from the professionals. The collected data provides a broader view of the main challenges and attributes they consider essential. The questions from the form, provided through Google Forms, are compiled in Table 3.

The professionals surveyed have a broad and varied experience in the field, as shown in Figure 2, with years of experience ranging from less than one year to over six years, as illustrated in Figure 3.

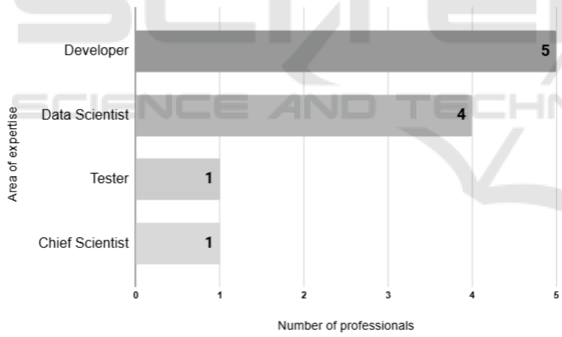

Figure 2: Professionals' Area of Expertise.

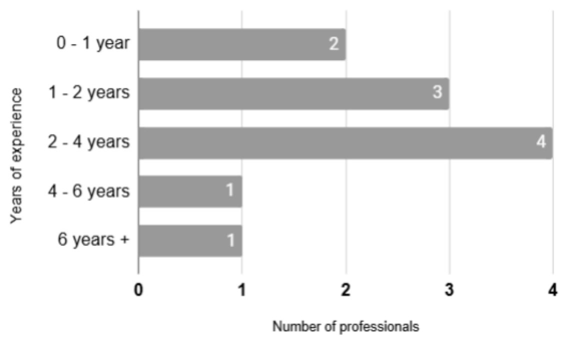

Figure 3: Years of Experience of the Professionals.

6.1 Data Quality Attributes

The primary data quality attributes identified by the professionals were consistency, accuracy, and relevance, as shown in Figure 4. These attributes were considered priorities because they ensure the collected data is complete, accurate, and relevant to business needs. Consistency ensures the data is uniform and coherent across different sources and systems. Accuracy ensures that the data is free from errors and inaccuracies, reflecting reality precisely. Relevance ensures the data is pertinent and valuable for the organization's objectives, enabling informed and effective decision-making.

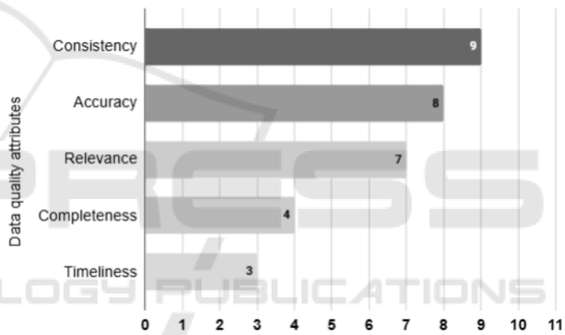

Figure 4: Most Relevant Attributes.

6.2 Identified Challenges

The challenges most frequently mentioned regarding data quality in Big Data systems included the need for more specific tests, such as load, performance, and integrity tests. These tests are necessary to ensure that the system can efficiently handle large volumes of data without compromising the performance or accuracy of the processed data. The absence of these tests can lead to significant failures, impacting the system's reliability and effectiveness.

Additionally, the need for better communication and integration between the development teams and the Product Owners (POs) was highlighted to ensure that data quality criteria are adequately met.

6.3 Testing Techniques

Among the testing techniques mentioned as missing, load and performance testing stand out, as they are

Table 3: Questions from the survey applied.

Questions	Response Options
Have you worked with applications that handle large volumes of data (big data)?	Yes or No
What was your role in the big data project?	Developer, Data Scientist, Tester, Other
How would you rate your experience in the field?	0-1 year, 1–2 years, 2–4 years, 4–6 years, more than 6 years
Which of these data quality attributes do you consider most important? (choose only 3)	Completeness, Accuracy, Timeliness, Consistency, Relevance, Other...
Rank the 3 attributes selected above in order of importance (e.g., 1. Option A, 2. Option B, 3. Option C)	Open-ended question
Do you believe the Completeness of data was evaluated in System X? How do you suggest evaluating the Completeness of data in a Big Data system?	Open-ended question
Do you believe the Accuracy of data was evaluated in System X? How do you suggest evaluating the Accuracy of data in a Big Data system?	Open-ended question
Do you believe the Timeliness of data was evaluated in System X? How do you suggest evaluating the Timeliness of data in a Big Data system?	Open-ended question
Do you believe the Consistency of data was evaluated in System X? How do you suggest evaluating the Consistency of data in a Big Data system?	Open-ended question
Do you believe the Relevance of data was evaluated in System X? How do you suggest evaluating the Relevance of data in a Big Data system?	Open-ended question
In your experience with System X, did you feel the need for any specific type of testing? (Choose only 3)	Load Testing, Performance Testing, Regression Testing, Usability Testing, Unit Testing, Integration Testing, Security Testing, Other...
In your experience with System X, did you feel the need for any specific type of testing?	Open-ended question

essential for evaluating the system's ability to process large volumes of data efficiently, as shown in Figure 5. These tests help ensure that the system maintains proper performance under high data load. Furthermore, integrity tests are also relevant for verifying that data flows remain consistent and error-free throughout the different processing stages. Applying these techniques is crucial to ensure the robustness and reliability of the system in Big Data environments.

The survey conducted through the form provided greater clarity on the most relevant data quality attributes and the challenges faced by the system development and quality teams. Adopting specific testing techniques and improving communication between teams is essential for enhancing data quality and, consequently, the effectiveness of decisions based on that data. These measures not only elevate the quality of the final product but also optimize internal processes, resulting in a more robust and reliable system.

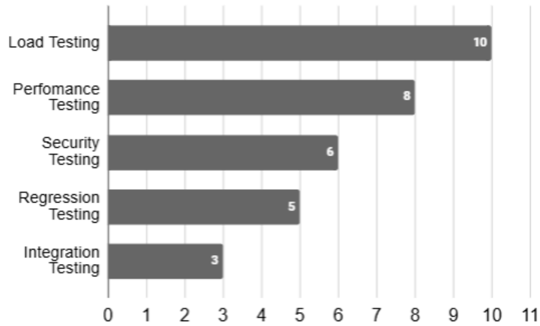

Figure 5: Most requested test types.

7 LESSONS LEARNED

During the development and testing of the Big Data Fortaleza platform, several valuable lessons were learned, significantly contributing to improving the development process and quality assurance. Here are some of the key lessons that emerged:

1. **Invest in Testing Processes to Ensure Data Quality.** Data quality is a crucial aspect of the success of a Big Data platform. During the system testing, it became evident that investing time and resources in ensuring data quality is necessary. This includes identifying, cleaning, standardizing formats, and ensuring data integrity and consistency, as poor data quality can compromise the effectiveness of analysis and decision-making. In general, it is observed that such activities are the responsibility of the data team, and there is no strong culture related to test coding.
2. **Need for Data Simulation.** An important lesson learned during the development and validation of the Big Data Fortaleza system was the need for data simulation, in addition to using accurate data, to effectively validate the dashboards present in the system. This approach allowed a more comprehensive verification of the dashboard functionalities, ensuring that they could handle a variety of scenarios and data volumes.
3. **Security Testing Should Be Conducted Continuously from the Start of the Project.** Data protection policies are crucial in Big Data projects, especially those dealing with sensitive data. This includes penetration testing, vulnerability analysis, data encryption, and restricted access policies to protect the integrity and confidentiality of the information stored and processed by the platform.
4. **Integration and Unit Tests Are Vital to Ensure the Reliability of Business Rules.** Since the system deals with sensitive data, conducting integration tests is crucial. These tests are designed to verify whether the different components of the system interact correctly with each other and with the database, ensuring the integrity and proper functioning of the application as a whole.
5. **Assertive Communication Between Teams Should Be a Priority.** Support and communication between developers and data scientists were essential for the testing team to perform their work effectively and contribute significantly to the success of the Big Data Fortaleza project. Constant communication with the development team allowed for continuous information exchange about bugs that should be prioritized, recurring failures, and knowledge transfer. Data scientists, in turn, helped the testing team better understand the data analysis requirements and identify possible inconsistencies. This collaboration resulted in a more comprehensive testing approach, ensuring early detection of issues and delivering a high-quality final product.

REFERENCES

Aydos, M., Aldan, Ç., Coşkun, E., and Soydan, A. (2022). Security testing of web applications: A systematic mapping of the literature. *Journal of King Saud University-Computer and Information Sciences*, 34(9):6775–6792.

Bourque, P. and Fairley, R. (2014). SWEBOK: Guide to the software engineering body of knowledge. IEEE Computer Society, Los Alamitos, CA, version 3.0 edition.

Costa, A., Freitas, L., Cavalcante, D., Oliveira, V., Lelli, V., Santos, I., Oliveira, P., Nogueira, T., and Andrade, R. (2024). Especificação de requisitos em um projeto de big data no setor público. In *Anais do XXVII Congresso Ibero-Americano em Engenharia de Software*, pages 417–420, Porto Alegre, RS, Brasil. SBC.

Daase, C., Staegemann, D., and Turowski, K. (2024). Overcoming the complexity of quality assurance for big data systems: An examination of testing methods. In *IoTBDS*, pages 358–369, Magdeburg, Germany. Institute of Technical and Business Information Systems.

de Oliveira, I., Lima, J. M., Cristhian, S., Santos, I. S., and Andrade, R. (2024). Quality of big data systems: a systematic review of practices methods and tools. In *SBQS 2024 - Trilha de Trabalhos Técnicos*.

Nasir, N., Imtiaz, S., Imtiaz, S., and Nabeel, M. (2022). Testing framework for big data: A case study of telecom sector of pakistan.

OWASP (2023). Owasp testing guide.

Punn, N. S., Agarwal, S., Syafrullah, M., and Adiyarta, K. (2019). Testing big data application. In *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE.

Rivero, L., Diniz, J., Silva, G., Borralho, G., Braz, G., Paiva, A., Alves, E., and Oliveira, M. (2020). Deployment of a machine learning system for predicting lawsuits against power companies: Lessons learned from an agile testing experience for improving software quality. In *Anais do XIX Simpósio Brasileiro de Qualidade de Software*, pages 294–303, Porto Alegre, RS, Brasil. SBC.

Santos, I., Oliveira, P., Oliveira, V., Nogueira, T., Dantas, A., Menescal, L., Élcio Batista, and Andrade, R. (2023). Big data fortaleza: Plataforma inteligente para políticas públicas baseadas em evidências. In *Anais do XI Workshop de Computação Aplicada em Governo Eletrônico*, pages 200–211, Porto Alegre, RS, Brasil. SBC.

Schwaber, K. and Sutherland, J. (2020). The Scrum Guide – the definitive guide to scrum: The rules of the game.

Staegemann, D., Volk, M., Daase, C., and Turowski, K. (2020). Discussing relations between dynamic business environments and big data analytics. *Complex Systems Informatics and Modeling Quarterly*, (23):58–82.

Taleb, I., El Kassabi, H. T., Serhani, M. A., Dssouli, R., and Bouhaddioui, C. (2016). Big data quality: A quality dimensions evaluation. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 759–765. IEEE.

Wang, J., Liu, Y., Li, P., Lin, Z., Sindakis, S., and Aggarwal, S. (2023). Overview of data quality: Examining the dimensions, antecedents, and impacts of data quality. *Journal of the Knowledge Economy*, pages 1–20.

Washburn, D., Sindhu, U., Balaouras, S., Dines, R., Hayes, N., and Nelson, L. (2010). Helping CIOs Understand "Smart City" Initiatives: Defining the smart city, its drivers, and the role of the cio. Cambridge, MA: Forrester Research.

Élcio Batista, Andrade, R., Santos, I., Nogueira, T., Oliveira, P., Lelli, V., and Oliveira, V. (2024). Fortaleza city hall strategic planning based on data analysis and forecasting. In *Anais do XXVII Congresso Ibero-Americano em Engenharia de Software*, pages 433–436, Porto Alegre, RS, Brasil. SBC.