Quality Assurance for Big Data Application– Issues, Challenges, and Needs

Chuanqi Tao
Computer Science&Engineering Department
Nanjing University of Science and Technology
Nanjing, China
taochuanqi@njust.edu.cn

Jerry Gao
Computer Engineering Department
San Jose State University, San Jose, USA
Taiyuan University of Technology, Taiyuan, China
Corresponding to: jerry.gao@sjsu.edu

Abstract**—**With the fast advance of big data technology and analytics solutions, building high-quality big data computing services in different application domains is becoming a very hot research and application topic among academic and industry communities, and government agencies. Therefore, big data based applications are widely-used currently, such as recommendation, predication, and decision system. Nevertheless, there are increasing quality problems resulting in erroneous testing costs in enterprises and businesses. Current research work seldom discusses how to effectively validate big data applications to assure system quality. This paper focuses on big data system validation and quality assurance, and includes informative discussions about essential quality parameters, primary focuses, and validation process. Moreover, the paper discusses potential testing methods for big data application systems. Furthermore, the primary issues, challenges, and needs in testing big data application are presented.

Keywords— Quality assurance, big dataapplication quality assurance, big data validation.

I. Introduction

According to IDC [1], the Big Data technology market will "grow at a 27% compound annual growth rate (CAGR) to $32.4 billion through 2017".Today, with the fast advance of big data science and analytics technologies, diverse data mining solutions, machine learning algorithms, open-source platforms & tools, and big data database technologies have been developed, and become available to be used for big data applications. This suggests that big data computing and application services bring large-scale business requirements and demands in people's daily life. Big data-based application system is widely-used nowadays, such as recommendation system, predictions, recognized patterns, statistical report applications, etc. Emergent big data computing and services can be used in many disciplines and diverse applications, including business management, library science, energy and environment, education, biomedical, healthcare and life science, social media and networking, smart city and travel, and transportation, etc.[2]. Nevertheless, due to the huge volume of generated data, the fast velocity of arriving data, and the large variety of heterogeneous data, the big data based applications brings new challenges and issues for QA engineers. For instance, it is a hard job to validate the correctness of a big data-based prediction system due to the large scale data size and the feature of timeliness. Therefore, Big data quality validation and big data-based application system quality assurance becomes a critical concern and research subject. Although there has been a numerous of published papers [2-6] addressing data quality and data quality assurance in the past, seldom researches focus on validation for big data application quality. There is an emergent need in research work to quality study issues and quality assurance solutions for big data applications.

This paper is written to provide our perspective views on big data system validation for quality assurance. The paper is organized as follows. Section II discusses the typical types of big data systems and covers the essential quality parameters and their associated factors. Section III reviews and compares the existing testing methods for big data system validation. The major issues, challenges, and needs are presented in Section IV. Conclusions are in Section V.

II. Understanding Quality Assurance for Big Data Application System

This section discusses the scope and process of quality assurance for big data application systems. Moreover, it covers the primary quality parameters with related factors.

Big data applications have the following unique features: (a) statistical computation based on multi-dimensional large-scale data sets, b) machine-learning and knowledge based system evolution, c) intelligent decision making with uncertainty, d) non-oracle functions, and e) complicated visualization. These unique features bring more interesting quality assurance and QoS requirements, challenges, and needs. Based on the recent feedbacks from engineers at Silicon Valley, how to assure the quality of big data-based application systems becomes a critical concern and research subject currently.

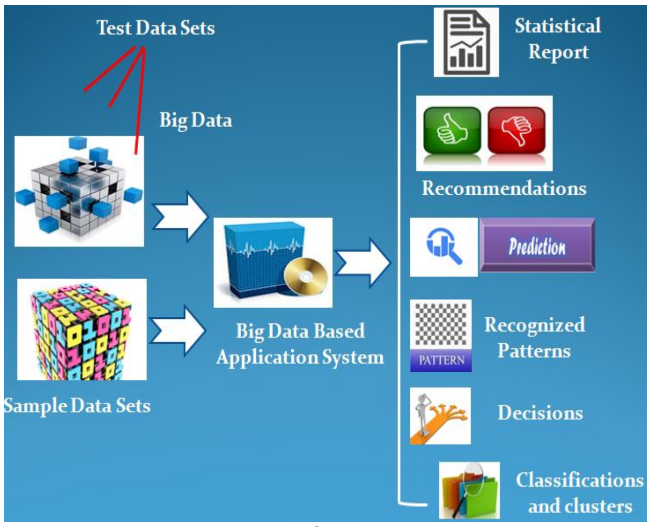

Figure 1 The Typical Types of Big Data Application Systems

DOI reference number: 10.18293/SEKE2016-166

A. *Scope and Process of Big Data Application Quality Assurance*

Big data applications provide services for prediction, recommendations, decisions support through large-scale data sets and complicated intelligent algorithms. Figure 1 describes the typical types of big data applications.

In general, ***big data application quality assurance*** refers to the study and application of various assurance processes, methods, standards, criteria, and systems to ensure the quality of big data system in terms of a set of quality parameters. Figure 2 shows a sample scope of validation for quality assurance of big data applications.

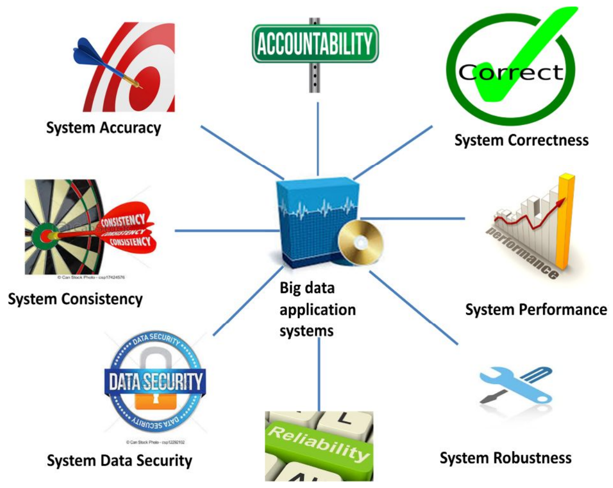

Figure 2 The Scope of Validation for Big Data Application System Quality

Compared to conventional software testing, a test process of big data based applications primarily focuses on their unique features, such as oracle problems, learning capability, and timeliness testing. Figure 3 shows a sample test process (function testing) for big data application system validation.

The testing process, shown in Figure 3, includes the following steps.

Step1 Big data system function testing, including rich oracles, intelligent algorithms, learning capability, as well as domain-specific functions;

Step 2 Big data system function testing, includingsystem consistency, security, robustness, and QoS;

Step 3 Big data system feature testing, checks usability, system evolution, visualization, and so on;

Step 4 Big data system timeliness testing, targets time related feature testing, including continuous testing, real-time testing, life-time testing, and others.

B. *Quality factors for big data application validation*

Conventional system quality parameters such as performance, robustness, security, etc., can be applicable onto big data systems. They are listed below.

- ***System Performance*** **–**This parameter indicates the performance of the system, such as availability, response time, throughout, scalability, etc.

- ***System Data Security*** **–**This parameter could be used to evaluate the security of big data based systemin different perspectives. Using this parameter**,** data security could be evaluated in various perspectives at the different levels.
- ***System Reliability*** –This parameter is used to evaluatethe durability of the system when performing a required function under stated conditions for a specified period of time.
- ***System Robustness*** **-** This parameter evaluates the ability of a system to resist change without adapting its initial stable configuration.

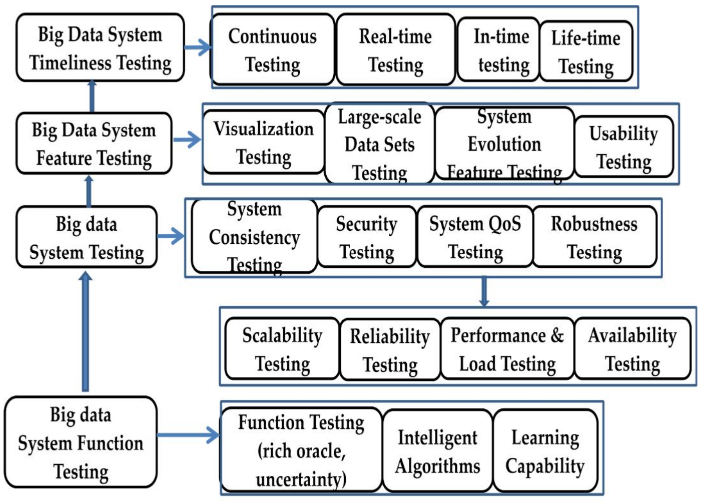

Figure 3 A Quality Test Process for Big Data Application System

In addition, due to the special characteristics, such as oracle problems, big data applications bring some impactedfactors contributing to the challenges of system quality assurance. There are two typical big data applications: a) recommendation systems, and b) prediction systems. We have collected a number of common quality parameters from survey. Figure 4 summarizes the typical quality factors for prediction and recommendation systems in a fishbone graph respectively. Those factors are presented in taxonomy below.

Quality factors for prediction systems

- ***System Correctness, which is a quality*** factor used to evaluate the correctness of the big data applications. Unlike the conventional system, big data applications are hard to validate their correctness. For instance, prediction–related software is mainly developed to make predictions or better understand about real world activities. Hence, it is difficult to determine the correct output for those types of software. Correctness is related to the prediction pattern or model. For instance, some models are more likely used to predict point of inflexion values while some other models are doing well in predicting continuity. Thus, in order to verify the correctness of the system effectively, engineers need to evaluate the capability of prediction in the specified conditions and environments.

- ***System Accuracy, which*** is used to evaluate if the system yields true (no systematic errors), and consistent (no random errors) results. Some big data applications are developed to find previously unknown answers, thereby only approximate solutions might be available. This can be called uncontrollable prediction. Some prediction is used to prevent something happening in the future, and the prediction result will affect actions or behaviors. In turn, those actions can promote the prediction result.

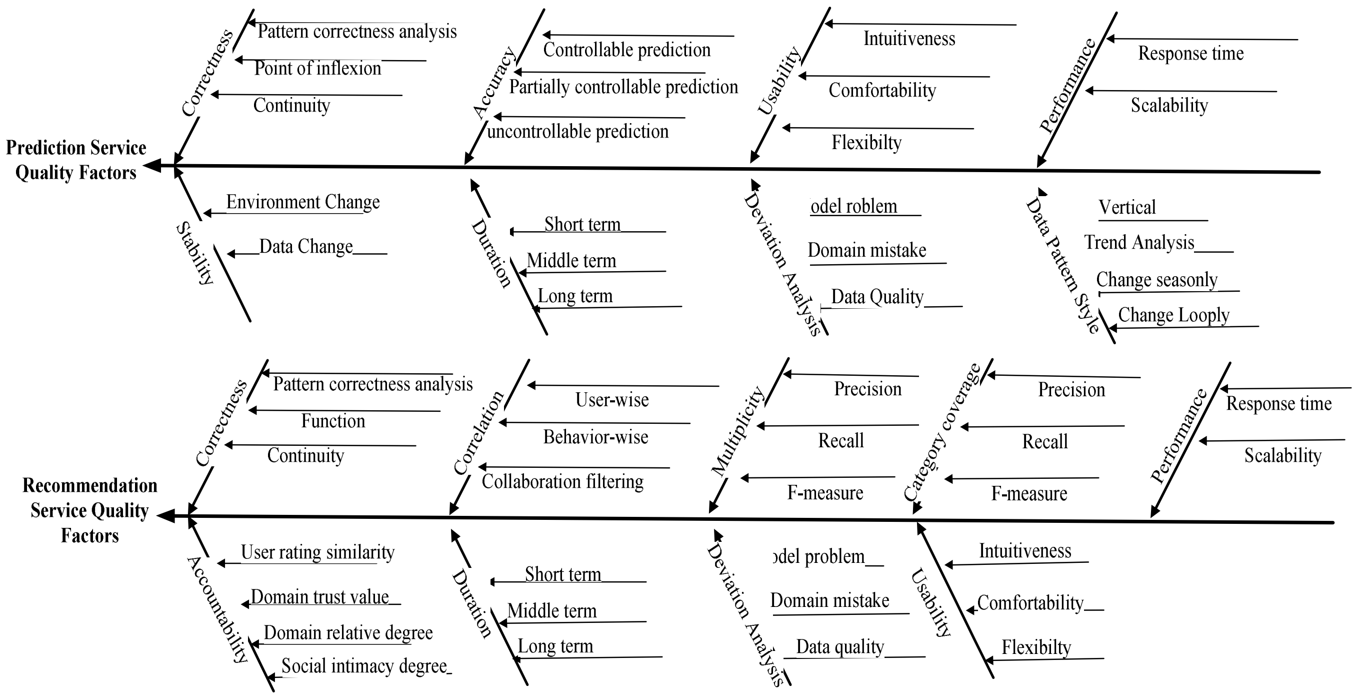

Figure 4 Big Data Application System Quality Factors

- ***System Stability, which*** reflects the stability of the system prediction while environment change or data changes. For example, if the prediction capability of a system is stable with little changes when statistical data are acquired from different timeframes.
- ***System Consistency, which is*** a quality indicator useful to evaluate the consistency of the targeted systemin different perspectives. Due to the inherent uncertainties in system models, some applications do not produce single correct output for a given set on inputs. This leads to hardly determining the expected behaviors of the software. In such situation, domain-specific experts could provide opinions to support system consistency.
- ***Duration, which*** indicates the expected prediction period. It can measure how up-to-date data is, and whether it is correct despite the possibility of modifications or changes that impact time and date values [6]. For instance, commonly-used prediction duration in enterprise management can be divided into short term, middle term, and long term.
- ***Deviation Analysis, which*** is used to analyze the prediction deviation within an accepted range or confidence interval.
- ***System usability, which*** is a parameter that indicates how well the big data application service can be used. This can be very subjective due to different developers and users have diverse user experiences. The typical usability factors include intuitiveness, comfortability, and flexibility.
- ***System Performance, which*** is a distinct quality factor for big data application service. It is useful to evaluate how well big data are structured, designed, collected, generated, stored, and managed to support large-scale prediction services.

Quality factors for recommendation systems

- ***Correctness*** **–**This quality factor reflects if the recommended service or commodity meets the demands of customers. Correctness could be subjective between different persons. Thus, how to measure correctness is still a challenge for quality assurance engineers.
- ***Correlation*** **–** This quality factor evaluates the degree of correlation of the recommended service. This involves various recommendation strategies, such as user content-based, behavior-based, and collaboration filtering-based.
- ***Multiplicity*** **–** This quality factor refers to the measurements for repeatability of recommended service. For instance, a poor quality system probably recommends too many repeated or similar commodities to users.

- ***Category Coverage –*** This indicator is useful to evaluate the coverage rate for diverse categories. This factor measures the completeness of recommendation within a selected domain.
- ***Accountability*** **–**This quality parameter is very important and mandatory for both big data service applications and users. This could be measured in a quantitative way, such as user rating similarity, domain trust value, domain related degree, and social intimacy degree.
- ***Duration*** **–**This factor indicates the expected recommendation period. For instance, commonly-used recommendation duration in enterprise management can be divided into short term, middle term, and long term.
- ***Deviation Analysis*** **–**This factor is used to analyze the recommendation deviation within accepted range or confidence interval.
- ***System usability*** **–**This parameter indicateshow well big data application service can be used. This can be very subjective due to different developers and users have diverse user experiences.
- ***System Performance*****–**This is a distinct quality factor for big data application service, and it is useful to evaluate how well big data are structured, designed, collected, generated, stored, and managed to support large-scale recommendation services.

In addition to the two typical applications discussed above, there are more big data related applications such as machine learning system, ranking system, and search system. Due to page limits, we do not list their quality factors here. A comparison of conventional testing and big data application testing in detail is presented in Table 1.

Table 1 A comparison of conventional testing and big data application testing

	Conventional Testing	**Big Data Application Testing**
Primary Objectives	- Validate the quality of software, including functions, programs, performance, etc.	- Provide on-demand testing services for big data application systems to support software validation and quality engineering process.
Testing Focuses	- Diverse software errors in its structures, functions, behaviors, user interfaces, and connections to the external systems. - System non-functional requirements such as performances, reliability, availability, vertical scalability, security, and etc.	- Non-oracle problem or rich oracle function problem. - Complicated algorithms. - Large-scale data input. - Complicated data models and integrations.
Test Input	- Limited scale - Specified data formats - Structured data	- Large-scale data volume with diverse formats and media - Structured and non-structured data - Timeliness
Testing Execution	- Offline testing in a test lab before product delivery. - Testing in a cloud-based test environment.	- On-demand test execution in a cloud-based virtual test environment. - Continuous testing for big data applications.
Test Coverage	- Function-based, data flow, structure-based, state diagram.	- To be developed; lack able currently.
Data Model	- Data partition, boundary analysis, etc.	- Training data; sampling data; classifier; image data classification; pseudo-oracles
Testing Environment	- A pre-configured test environment in a test labwith purchased hardware/software and tools.	- Provide testing environment references - Develop rapid reuse framework
Testing Process	- Enterprise-oriented test processes for each project.	- Crowd sourcing-based process - Learning-based testing - Classification based testing
Testing Techniques	- Apply selected well-known white-box and black-box testing techniques at the component level (or unit level) and the system level.	- Required innovative continuous, timeliness, and currency testing techniques. - New testing solutions to deal with multi-dimensional large-scale data sets, uncertainty data, learning-based system evolution, and complicated visualization.
Testing Tools	- Use limited testing solutions and tools with the purchased licenses. - Select and use diverse testing tools solutions which are pre-configured, installed, and deployed.	- Support development process. - Construct the whole process tool chains. - Data analysis tool. - Continuous evaluation including crowdsourcing and sampling.
Tool Connectivity and Platform	- Traditional test tool/solution integration and composition.	- Domain-specific application. - On-demand selective solutions which support users to integrate and composite test solutions and tools. - Incremental data sets.

III. VALIDATION METHODS FOR BIG DATA APPLICATION

This section discusses and reviewsthe existing research results in software testing methods which have been used to validate various types of big data applications including intelligent systems, data mining programs, bioinformatics programs, and learning based applications.

Program-based software testing –Conventional program-based testing methods have been used in big data analytics applications. Csallner et al. presents a novel technique that systematicallysearches for such bugs in MapReduce applications and generates corresponding test cases [18]. The technique works by encoding the high-level MapReduce correctness conditions as symbolic program constraints and checking them for the program under test. Shang et al. proposed an approach to uncover the different behaviors of the underlying platforms for BDA Apps using Handoop between runs with small testing data and large real-life data in a cloud environment [19].

Classification-based testing-A classification approach to program testing usually involves two steps: a) training a classifier to distinguish failures from successful cases on a selected subset of results, and then b) applying the trained classifier to identify failures in the main set of results. A resembling reference model is usually used to train a classifier. More specifically, there are techniques for applying pattern classifications to alleviate the test oracle problems. Last et al. [9] and Vanmali et al. [11] apply a data mining approach to augment the incomplete specification of legacy systems. They train classifiers to learn the casual input-output relationships of a legacy system. Podgurski et al. classify failure cases into categories [10]. However, they do not study how to distinguish correct and failure behaviors of programs. Later, their research group further proposes classification tree approaches to refine the results obtained from classifiers [8]. Bowring et al. use a progressive machine learning approach to train a classifier on different software behaviors [7].They apply their technique in the regression testing of a consecutive sequence of minor revisions of a program.

Metamorphic testing (MT) - This is a classic approach to testing programs that do not have oracles. Whenever a formal oracle is not available or costly to apply, we run into a *test oracle problem*. A test oracle is a mechanism against which testers can check the output of a program and decide whether it is correct. When an oracle is not available, other means of determining whether the test result is correct are known as *pseudo-oracles*. MT operates by checking whether aprogram under test behaves according to an expected set of properties known as metamorphic relations. A metamorphic relation specifies how a particular change to the input of the program should change the output. MT was used for testing scientific applications in different areas such as machine learning applications [12, 13], bioinformatics programs [14], programs solving partial differential equations [15] and image processing applications [16]. When testing programs solving partial differential equations, MT uncovered faults that cannot be uncovered by special value testing [15].

Learning-based testing – This involves how to adopt the various learning approaches and mechanisms to support testing for big data systems. Meinke et al. developed a technique for automatic test casegeneration for numerical software based on learning based testing (LBT) [17]. The authors first created a polynomial model as an abstraction of the program under test. Then the test cases are generated by applying a satisfiability algorithm to the learned model.

Crowd-sourced testing–This testing approach uses freelance testers and/or contracted engineers in a crowd sourcing community. It is a cost-effective method to validate a machine-learning based application systems, such as a human face recognition system. Currently, crowd-sourced testing has been used in mobile app testing and mobile TaaS (Testing as a Service). One good example is uTest (http://www.utest.com/company).

Data model-based testing –Since big data are the input values for big data application systems, diverse data models can be used to assist test case generations. Vilkomir et al. presents a method to automatically generate test cases for a scientific program having many input parameters with dependencies [20]. They use a directed graph to model the input data space, including parameters and values as well as their dependencies. Valid test cases can be automatically generated based on the directed graph model. Since their model satisfies the probability law of Markov chains, it can be used to generate random and weighted test cases according to the likelihood of taking the parameter values.

Rule-based software testing –This approach could be used in testing rule-based or knowledge based systems. The basic idea is to design test cases based on the rules specified in an expert system. Deason et al. in [21] proposed a rule-based test data generation method for Ada programs. They demonstrated that rule-based test data generation is feasible. The paper shows a great promise in assisting test engineers in test generation. Andrews et al. presented a test pattern generation approach based on VHDL specific heuristic rules [22]. Their results indicated the rule-based approach leading to a better test coverage.

Conventional black-box software testing –To assure the system performance and other related QoS parameters of big data applications, engineers could use convention black-box approaches to controlling their quality. Typical examples include decision table testing, equivalence partitioning, boundary value analysis, cause-effect graph, and use case testing, and so on.

IV. ISSUES, CHALLENGES, AND NEEDS

There are a number of major issues and challenges in big data quality validation and assurance. Here are typical ones.

Issue #1*–What are the adequate test models and test coveragecriteria for big data based service applications?*

With the fast advance of big data technologies and analytics methods, more and more big data based applications and service systems are developed to be used in many areas of our daily life, including smart cars, smart city, business intelligence, environmental control, and so on. The increasing deployment of big data applications and services raises quality assurance concerns. In the past, many existing white-box and black-box software test models and adequate validation criteria are developed to address validation needs of software applications in functions, behaviors, and structures. However, these existing adequate test models only focus on program functions, state-based behaviors, and program structures. In the software testing and quality assurance community, there is a lack of research work on adequate test modeling and coverage analysis for big data application systems by considering their special features and needs in rich oracle functions, machine-learning based system evolutions, knowledge based system intelligence, and multi-dimensional large-scale data sets.

Hence, according to real world practitioners, there is a clear demand on establishing well-defined test coverage criteria for big data application systems. Otherwise, engineers and big data analysts will have difficult time to figure out when they should stop quality testing for big data applications. This leads to the first demand described below.

Need #1*–Developing well-defined adequate validation models and criteria to address the special features and needs of big data applications and services.*

Issue #2 *–Where are the well-definedbig data system quality assurance programs and standards, including processes, assessment metrics, regulations, and policies?*

As we discussed in [25], ISO is working on updating of existing data quality assurance standards and programs for big data quality assurance. Considering the popularity of big data applications and services, we must address the quality control and assurance of big data based applications. Here, we point out the second emergent need below.

Need #2 *– Establishing quality assurance programs and standards to consider the special QoS parameters and factors of big data applications and services to ensure system quality.*

Issue #3*– What are test automation solutions and tools supporting efficient and large-scale testing operations for big data applications and services?*

In the past three decades, many test automation tools and solutions have been developed for engineers in assisting test automation activities and operations. Unfortunately, most of these tools are only useful to validate software system functions, behaviors, program structures, and system performance and other QoS parameters. As discussed in section III, there are a few published papers addressing validation methods.

However, these validation methods are not designed and developed to address the special features and needs of big data application systems. As discussed in Section III, there has been a few of published research work addressing special validation needs in big data based applications. However, there is a clear lack of research work on automatic validation methods and solutions for big data application services. Therefore, the third emergent need for big data applications is listed below.

Need #3 *–More innovative adequate testing methods and test automation tools to address the special needs and features of big data application systems and services.*

Unlike conventional software test automation tools, these expected test automation solutions must consider big data applications' special features listed below:

- Large-scale big data inputs with diverse formats, and structured and non-structured data;
- Learning and knowledge based system evolutions;
- Non-oracles problems and rich oracle functions with uncertainty;
- New QoS parameters, such as accuracy, accountability, usability, and
- Data modeling

V. CONCLUSIONS

With the fast advance of big data management technologies and analytics solutions, how to build high-quality big data application services becomes a very hot subject. Nevertheless, there are increasing quality problems resulting in erroneous data costs in enterprises and businesses [25]. Current research work seldom discusses how to effectively validate big data applications to ensure system quality. This paper provides informative discussions on big data system validation and quality assurance, including the essential concepts, focuses, and validation process. Moreover, the paper identifies and discusses some primary quality factors. In addition, it presents a comparison between conventional testing and big data application testing. Furthermore, the primary issues, challenges, and needs are presented.

REFERENCES

[1] Editor of Hosting Journalist, IDC forecast: big Data technology and services to hit $32.4 billion in 2017, December 18, 2013.

[2] A.O. Mohammed, S.A. Talab. Enhanced extraction clinical data technique to improve data quality in clinical data warehouse. International Journal of Database Theory and Application, 8(3): 333-342, 2015.

[3] M. R. Wigan, R. Clake. Big data's big unintended consequences. IEEE Computer, 46 (6):46-53, 2013.

[4] J. Alferes, P.Poirier, C. Lamaire-Chad, et al. Data quality assurance in monitoring of wastewater quality: Univariate on-line and off-line methods. In Proc. of the11th IWA conference on instrumentation control and automation, pp. 18-20, September, 2013.

[5] R. Clarke. Quality factors in big data and big data analytics. Xamax Consultancy Pry Ltd. 2014.

[6] A. Immonen, P. Paakkonen, and E. Ovaska. Evaluating the quality of social media data in big data architecture. IEEE Access, 3: 2028 - 2043 October 16, 2015.

[7] J.F. Bowring, J.M. Rehg, and M.J. Harrold. Active learning forautomatic classification of software behavior. InProc. of the 2004 ACM SIGSOFT International Symposium on SoftwareTesting and Analysis (ISSTA). ACM, New York, NY, pp.195–205, 2004.

[8] P. Francis, D. Leon, M. Minch, and A. Podgurski. Tree-basedmethods for classifying software failures. In Proc. of the 15th International Symposium on Software Reliability Engineering (ISSRE). Los Alamitos, CA, pp. 451–462. 2004.

[9] M. Last, M. Friedman,and A. Kandel. The data mining approachto automated software testing. In Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery andData Mining (KDD). ACM, New York, NY, pp. 388–396, 2003.

[10] A. Podgurski, D. Leon, P. Francis, W. Masri, M. Minch, J. Sun, and B. Wang. Automated support for classifying software failure reports. In Proc. of the 25th International Conference on Software Engineering (ICSE), LosAlamitos, CA, pp. 465–475, 2003.

[11] M. Vanmali, M. Last, and A. Kandel. Using a neural networkin the software testing process. International Journal of Intelligent Systems, 17 (1): 45–62, 2002.

[12] X. Xie, J.W. Ho, C. Murphy, G. Kaiser, B. Xu, and T.Y. Chen. Testing and validating machine learning classifiers by metamorphic testing. Journal of System and Software, 84 (4):544−558, 2011.

[13] C. Murphy, G. Kaiser, L. Hu, and L. Wu, Properties of machine learning applicationsfor use in metamorphic testing. In Proc. of the 20th International Conferenceon Software Engineering and Knowledge Engineering (SEKE), pp.867−872, 2008.

[14] T.Y. Chen, J.W.K. Ho, H. Liu, and X. Xie. An innovative approach for testing bioinformatics programs using metamorphic testing, BMC Bioinform. 10(2009).

[15] T. Chen, J. Feng, and T.H. Tse. Metamorphic testing of programs on partial differential equations: a case study. In Proc. of the 26th Annual International Computer Software and Applications Conference, (COMPSAC), pp. 327–333, 2002.

[16] J. Mayer, R. Guderlei. On random testing of image processing applications, In Proc. of the 6th International Conference on Quality Software (QSIC), pp. 85−92, 2006.

[17] K. Meinke, F. Niu. A learning-based approach to unit testing of numerical software, in: A. Petrenko, A. Simo, J. Maldonado (Eds.), Testing Software andSystems, Lecture Notes in Computer Science, vol. 6435, Springer, Berlin,Heidelberg, 2010, pp. 221−235.

[18] C. Csallner, L. Fegaras, C. Li. New Ideas Track: Testing MapReduce-style programs. In Prof. of 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE’11), pp 1-4, 2011.

[19] W. Shang，Z.M. Jiang，H. Hemmati，B. Adams，and A.E. Hassan. Assisting developers of big data analytics applications when deploying on Hadoop clouds. In Prof. of 35th International Conference on Software Engineering (ICSE), pp 402-411, 2013.

[20] S.A. Vilkomir, W.T. Swain, J.H. Poore, and K.T. Clarno. Modeling input space fortesting scientific computational software: a case study. In Prof. of the 8th International Conference on Computational Science, Part III (ICCS), pp. 291−300, 2008.

[21] W.H. Deason，D.B. Brown，and K.H. Chang. A rule-based software test data generator. IEEE Transactions on Knowledge and Data Engineering, 3(1): 108-117, 1991.

[22] A. Andrews, A.O. Fallon, and T. Chen. A Rule-Based Software Testing Method for VHDL Models. VLSI-SOC 2003: 92.

[23] W. Afzal, R. Torkar, and R. Feldt. A systematic review of search-based testing for non-functional system properties. Information and Software Technology, 51 (6):957–976, 2009.

[24] T. Clune, R. Rood, Software testing and verification in climate model development, IEEE Software, 28 (6):49–55, 2011.

[25] J. Gao, C.L. Xie, and C.Q. Tao. Quality assurance for big data–issuses, challenges, and needs. In Prof. of IEEE 9th International Symposium on Service oriented System Engineering, OZFORD, UK, 2016.

ACKNOWLEDGEMENT

This paper is supported by the National Natural Science Foundation of China under Grant No.61402229 and No.61502233; the Open Fund of the State Key Laboratory for Novel Software Technology (KFKT2015B10), and the Postdoctoral Fund of Jiangsu Province under Grant No.1401043B.
.