WHITEPAPER

Agile Big Data: Efficient Quality Management of Functional and Non-functional Characteristics

sqs.com

Authors: Michael Recktenwald
Service Lead Big Data (DACH)

SQS Software Quality Systems Switzerland

Helmut Körfer
Local Head of Service Test Data (DACH)

SQS Software Quality Systems Germany

Published: September 2016

Transforming the World Through Quality

MICHAEL RECKTENWALD

Service Lead Big Data (DACH)
michael.recktenwald@sqs.com

Michael Recktenwald has been active in quality management for 15 years and has been a consultant at SQS since 2008. As a certified test manager and scrum master he specialises in E2E testing and DWH testing. Since 2015, he has been responsible as Service Lead Big Data DACH for Big Data-related quality services.

HELMUT KÖRFER

Local Head of Service Test Data (DACH)
helmut.koerfer@sqs.com

Helmut Körfer has been a consultant at SQS AG since 2011, during which time he has introduced an international IT asset management system within the SQS Group. Since 2012 he has been a specialist consultant for test data management and test environment management, as well as being responsible for the issue as local head of service test data for service development in DACH.

Contents

Management summary 4
Keywords 4
Market analysis 4
Efficient quality management of Big Data system requirements 5
Phases of a Big Data system 6
Influence on methods of quality assurance 7
Influence on non-functional testing 8
Influence on test management 9
Continuous automated validation 10
Influence on test strategy 10
Conclusion and outlook 12
References 12

Management summary

For reasons of data protection, data abuse and potentially erroneous use in the era of predictive analytics, the requirements and expectations regarding the security and quality of Big Data systems have very much come to the fore. At the same time, however, Big Data projects and the associated systems are expected to deliver quick results and sufficient added value, which is why businesses often forego a Big Data test environment and confine themselves to building up a productive Big Data system [1]. However, we show that the Big Data approach makes sense and delivers benefits in terms of meeting quality and security requirements, provided it is linked with continuous verification and continuous validation. We go on to demonstrate that the quality expectations for Big Data systems can be met more effectively using agile methods rather than traditional ones. In this white paper we also explain what influence these changes have on the existing test disciplines, roles and test strategies.

Keywords

BIG DATA CONTINUOUS VERIFICATION CONTINUOUS VALIDATION DATA MANAGEMENT SERVICE VIRTUALISATION

Market analysis

One-third of the German companies surveyed in the Bitkom study already use Big Data solutions and more than two-thirds are open to and interested in the issue of Big Data [2]. We can expect this interest in Big Data to increase further over the coming years. This is also clear from the fact that one-quarter of those questioned plan to implement Big Data solutions in the future. However, the principal stumbling blocks for the companies surveyed are data protection and the resources needed.

Efficient quality management of Big Data system requirements

The terms verification and validation and how to use them are key to understanding the Big Data Quality Management concept described here. Verification is the process of checking the correct functioning of a system's components, whereas validation is the process of checking whether the results delivered by the system comply with the formal acceptance criteria [3].

In the analytical world and the world of Big Data, testing is divided into these two parts, firstly verification within the test environment and secondly validation within the production environment. Verification within the test environment is feasible using traditional methods. However, validation poses intractable problems for the user. The main reason for this is the need to test or validate an analytical system with live data and also under live conditions. That is the only way to ensure that the system is functioning correctly, or to correct erroneous data sets. If no more errors are detected, the analytical system is deemed error-free and the results generated can then be used unrestrictedly. However, this approach conceals a hidden danger. A user of a BI system cannot assess the data quality of the underlying report as it stands. Furthermore, it is often impossible for the user to tell whether the data has been correctly processed or not, and this can lead to errors not being detected over a long period. In Big Data systems this danger is exacerbated by the increased data volume, the sheer complexity and the use of algorithms. From a quality assurance viewpoint, this therefore poses the question of how a Big Data system can most effectively be tested in order to ensure that all the software quality requirements are met.

But what do we actually mean by Big Data systems? In a Big Data system large volumes of data are gathered and prepared with the aid of a Hadoop system landscape and its derivatives. Apache Hadoop is a framework in Java for scalable software, which functions in a distributed way. It is based on the MapReduce algorithm as well as recommendations from the Google file system, and it thereby facilitates intensive computing processes involving large data volumes on computer clusters [4].

In SQS' view the best way to test a Big Data system is by developing and testing it in phases (Table 1).

In our view, this procedure of building up a Big Data system structure over a number of phases can be most effectively executed with the aid of an agile development and a quality management process. This will allow you to most effectively pursue an iterative approach applying the 'start small, fail fast' principle. The method involves testing to arrive at rapid results which the agile team analyses and adjusts whenever its requirements are not met [5]. This assumption is supported by various studies, which have shown that interdisciplinary teams and the iterative process model approach are the key success factors for Big Data systems [6]. Similarly, in the case of Agile BI, Forrester Research has shown how agile methods deliver clear added value as compared with the traditional approach in the analytical world [7].

Phase	Name	Description of BDS	Team's test objectives
1	**Prototype**	Productive and non-productive data are read in and processed	Verification of the system and initial validation of the data design
2	**Pilot**	A pilot environment is produced from the prototype	Reading in productive data, possibly in combination with synthetic data for initial verification and validation
3	**New production**	Pilot environment goes live	Verification and validation of the results, to determine whether they are usable for decision-making
4	**Adjusted production**	Adjustments to the productive environment	Checking the changes and their results
5	**Production operation**	Continuous verification and validation of the system plus comparison with the decisions taken and their consequences	Assessing whether the probability statements from the Big Data results led to the right decisions being made

Table 1: Phases in a Big Data system and their testing objectives

Phases of a Big Data system

At Phase 1 the Big Data team can verify a system via manual or automated tests but cannot validate it. Alongside the non-functional requirements it must also check the data scientist's data design. In the process, all acceptance criteria are part of the 'definition of done' decided upon by the team. Therefore it is vital to involve the users of the Big Data system right from the start in order to adopt their acceptance criteria, because only if all requirements are met can the prototype be used productively. The team uses Phase 2 to test the requirements and then passes the prototype on to the pilot phase. The system will only be used productively during Phase 3 if validation and verification are successful. The question which arises in the process is whether all requirements have really been met. Even if further IT issues are resolved, for instance non-functional or regulatory requirements concerning data protection and data security, this raises the question as to whether this will remain the case, whether the results are correct and whether they can be applied by the user without misgivings. During the next change at the latest, that is in Phase 4, checks must again be made in order to ensure that all requirements are still being met. The team can only ensure that the system is continuing to do what it should do if the checks are carried out regularly. Furthermore, whether the results produced can be used for decision-making purposes cannot be answered because nobody can predict whether the forecast is accurate or not. Therefore it is important to check the consequences of the user's decisions. This is done during Phase 5. A consequence of this is that there is no longer any 'final GO', since the system is productive at all times, and the changes are checked directly as part of the production process.

In the SQS Agile Big Data testing process the quality assurance therefore takes place in parallel with the specification, development and implementation processes. Thus test-driven and security-driven development are combined. In the process the Big Data team must first specify what it wants to check and how it is going to do so, as well as simultaneously defining the potential risks and determining how to assess or reduce these risks. These checks are then integrated into the software so that they can be performed automatically. In addition, the necessary testing disciplines will be combined, reflecting the interdisciplinary nature of the project team. To render the results ready for release, they will first be subjected to additional statistical testing to check the probability statements. However, the key consideration for users is that all the decisions they make on the basis of these results must be fed back into the analysis of the system's behaviour. Assessing the decisions is very important because it is the only way to determine whether decisions had a negative effect or were based on a misinterpretation of the data. The challenge here will be to do this with sufficient frequency to allow countermeasures to be taken. Otherwise one would only have three options regarding the use of Big Data systems: Either the operators of the Big Data systems and the users of the outputs ignore this risk and use the system for as long as they gain a benefit from doing so, or they operate the systems until a third party identifies this risk and classifies it as too high. The third alternative is not to use the system.

Influence on methods of quality assurance

Alongside standards such as the ISO 29119 Software Testing Standard [8], the test methods used also play an important role in today's quality assurance. The aim here is to verify the specification in the test environment. In contrast, in the analytical world the final validation of the requirements often takes place during production or using productive data. For this purpose either a sandbox environment is used during production or the productive data are integrated in a preproduction process because the necessary test data are not available in the test environment on time and cost grounds. In Big Data systems it becomes even harder to make the necessary data available. Because of this, SQS supports testing throughout the individual system development phases via its SQS Big Data Management Services. These manage the data and ensure its availability in order to allow testing of the data volume, data variety, data velocity and data veracity, with the aim of ensuring that the fifth 'V', value, is delivered for the user. For Big Data, non-functional tests such as security tests, performance tests and data model verification are of key importance as they also cover the four Vs. Currently we identify non-functional tests such as the following: security tests, data consistency tests, stress tests, load tests, performance tests, etc. In Big Data systems these tests play a far greater role, if not the decisive role, in determining whether the system is working properly (Table 2).

Big Data requirement	Description	Test type
Volume	Data volume	Load test
Variety	Bandwidth of data types and data sources	Combination of tests
Velocity	Can all data be processed with the necessary speed?	Performance test
Veracity	Meaningfulness and trustworthiness	Statistical testing techniques

Table 2: Test types for Big Data requirement

One decisive point, whether the results we obtain from a Big Data system are correct, cannot be answered with complete certainty since this is a matter of correlations or probabilities. Despite this, a suitable verification process must be available when using a Big Data system. For this purpose data consistency and computer network tests are used. However, we would go so far as to define obtaining the proof necessary to permit the release of the results produced as an independent non-functional testing discipline. Firstly because the system has to be working correctly, that is, it must process the data properly in line with the definition. However, because the results have been independently interpreted by the system through the use of algorithms which also require testing, prior to release the results must also be tested using statistical methods. This will have to be done with the aid of statistical parameters and methods which define usability. This does not, however, give the users of the results carte blanche to take their decisions purely on the basis of those results, since a result obtained via algorithms is not absolute; rather it indicates tendencies. Just as nobody today would base decisions on extrapolations, nor should any decision-maker make decisions purely on the basis of Big Data results. Rather, methods must be devised whereby a correlation can be established between the reports, the decisions taken and the consequences of those decisions.

Influence on non-functional testing

Currently non-functional testing is normally only done to a limited extent on test environments. However, with Big Data systems these tests must be conducted continuously and automatically during all five phases as this is the only way to effectively safeguard the production environment. Moreover, to allow the checking of security and data protection requirements, the entire Big Data team must have access to the productive data, and this will undoubtedly pose problems for some organisations. If they do not have access, though, it would probably be difficult to guarantee that these requirements are met. The SQS solution supports businesses in meeting this new challenge by extending all testing disciplines in the existing catalogue of services so that they fit in with the Big Data service approach, thus facilitating the efficient testing of the non-functional areas in particular. The key factor here is to extend the various testing disciplines in a way which permits acceptance testing to be carried out in the production environment.

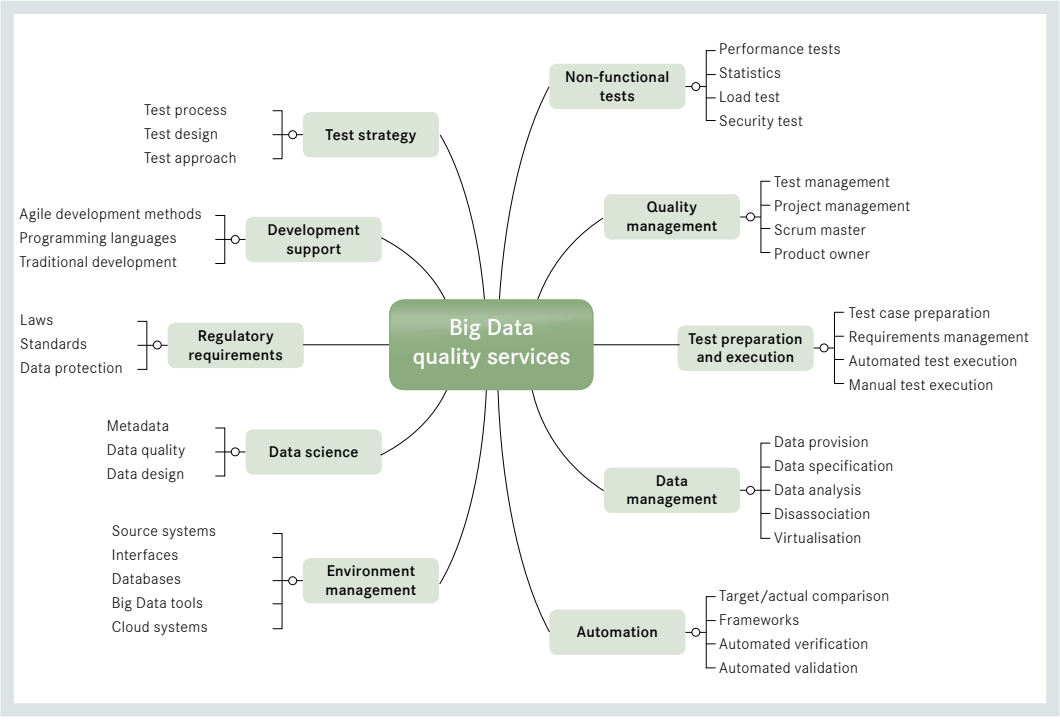

Figure 1: SQS Big Data quality services

To this end all the necessary SQS quality services depicted in Figure 1 are geared towards Big Data. They can all be used individually by the team, but they may also be deployed jointly in line with the interdisciplinary approach in order to cope with the Big Data complexity arising from the requirements of the five Vs.

Influence on test management

Currently test managers run the various tests in a test environment. They devise the testing strategy and have ultimate responsibility for making release possible without actually themselves being responsible for said release. They support stakeholders by providing the basis for their decision-making.

In a Big Data environment the test manager tends to assume the role within the team of a project leader, product owner or scrum master who coordinates the various testing activities. This becomes all the more the case given that the agile approach does not cater for the role of team test manager. Instead it applies the principle that the team monitors quality collectively. A traditional test manager would find this harder than a test manager with experience of agile or analytical methods. However, in the end, even a Big Data team needs a testing strategy in order to be able to monitor system quality throughout the individual phases. Moreover, calling in independent quality specialists offers the only possibility for building up and safeguarding confidence in a Big Data system.

Continuous automated validation

During phases 1 and 2 the prototype and pilot phase test activities may differ little from current methods, but in phases 3 to 5 quality assurance within the productive system plays the key role. The SQS Big Data Quality Service supports both verification and validation during the production process by combining and efficiently deploying the various testing disciplines, test automation, requirements engineering, test data management and test management. Initially the team concentrates on verification, checking that the functioning of the Big Data system is technically correct. However, to ensure that the process can be conducted continuously it will have to be automated. Similar to today in the analytical field, alongside verification the validation of the production requirements will play a far more important role. In Big Data systems, however, this is not done via the final release. Instead, continuous automated validation during the production process plays an important role as part of the requirements validation solution. The reason for this is that said release soon becomes obsolete because new results are continuously and automatically generated by the Big Data application, and these results then have to be validated.

In the end they can only be validated by checking the continuously produced output. There are no hard-and-fast requirements here, which could for instance be tested using an equivalence class or limit values as an example. Instead testing involves the deployment of statistical methods to analyse probabilities, correlations and tendencies. Accordingly it will be necessary to introduce statistical testing techniques. The aim here is not to demonstrate absolute correctness, but rather to determine whether the data design satisfies the statistical requirements. In other words, whether the algorithms are interacting correctly. The architecture and correct interaction of the various software components must also be tested, the final thing requiring testing is whether the results are usable and whether the decisions based on them generate added value and do not cause any loss or damage. The SQS Big Data Quality Service provides a variety of services, which facilitate the efficient execution of these tests and fulfilment of provability obligations.

Influence on test strategy

The current analytical test process, which is also recommended for Big Data, generally involves comparing the test results with a reference [9]. Operators are aiming for testing which follows a linear course whereby the master can be compared with an expected result. These often involve before & after comparisons or comparisons with the values required in the system to be delivered. This process of validation of content against a reference eventually leads to final release. However, in a Big Data environment it will be difficult not only to define a reference but also to arrive at final release. When developing a Big Data system it can only be compared with the values that the system itself has generated. Furthermore, this can only be done on the basis of verification, which checks whether all the data sets have been read in. The objective of a Big Data system testing strategy is to adhere as closely as possible to reality, because only then will a test be meaningful. However, due to the complexity of Big Data systems, it is difficult to closely model production processes when outside the productive system. The SQS Big Data Quality Service takes this state of affairs into account and provides assistance through data management and virtualisation. These two procedures deliver the required data in virtually the same form as they appear in the production process but with less effort.

This permits the early testing of a Big Data system if the data sources change (see Figure 2).

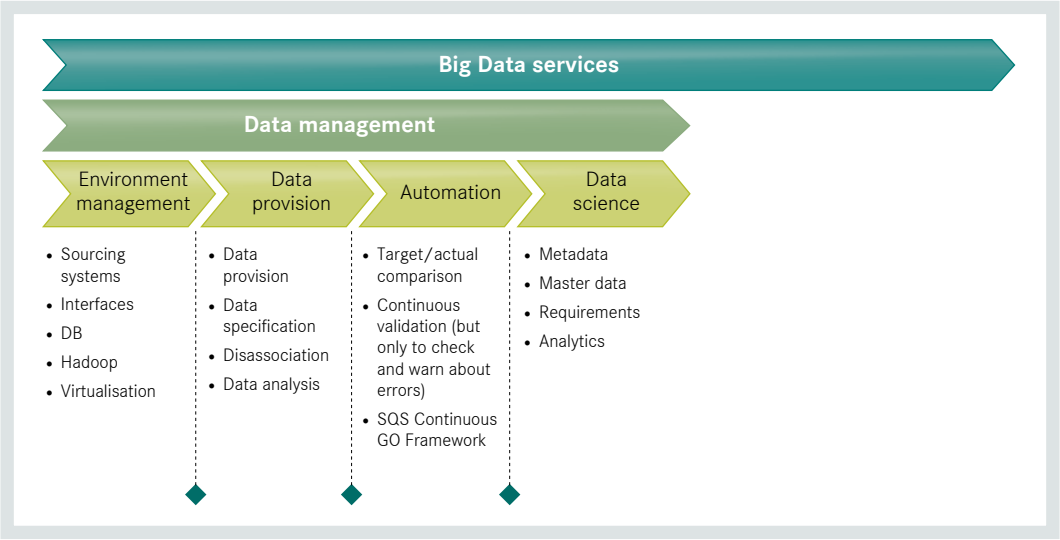

Figure 2: SQS data management

The SQS Continuous GO Framework, whereby verification and validation are performed continuously, will also be implemented. Currently the objective in the testing environment is to ensure that the productive processes continue to work after a software change and that results from the analytical system are not used until they have been checked and released. However, these two objectives are difficult to meet in a Big Data system because it is structured in such a way that it delivers results even if no individual data sources are present. This means that it will not be possible from the results alone to determine whether or not they are usable. Furthermore, the results produced by a Big Data system are never simply right or wrong. Rather, the sole criterion is how high the probability is that these results are right. Moreover, the results generated by a Big Data system only ever constitute a basis for interpretation. The user of the results always takes a decision after having obtained the results. Thus the more telling question is whether the user of the results, regardless of whether that user is human or a machine, has made the right decision.

In the case of a change to a productive Big Data system, this also means firstly that it is necessary to ensure that the data processing is still functioning correctly, and secondly that the results may only be used with the proviso that they are based on a modified data source. Here we have to make a distinction between static systems, such as in-house analysis systems, and mobile systems such as self-driving cars. In the first case it is easier for the user to determine whether the results can be used or not, whereas in the second case we have to ensure that the system is still behaving correctly. To this end it is necessary to have two productive systems available. In the changed system only certain modules are supplied with the new information and the actual productive system is only adjusted after successful validation. For all these reasons it is necessary to continuously monitor a Big Data system in order to operate it productively. This is the case firstly in order to document the transparency of the processes and comply with regulatory requirements, and secondly in order to provide as reliable as possible a basis for decision-making.

Conclusion and outlook

If we want to use Big Data systems, we must ensure that the quality management system is set up accordingly. In most cases manual testing will have to be replaced by automated verification and validation processes which are integrated into the software. However, although these may be good, the decisions made on the basis of Big Data results are what really matter. There will always be a residual risk that the wrong decisions will be made. When this happens, the important thing is to understand whether this happened because of false results or an incorrect interpretation of the results. Only then will we have the chance, in a Big Data environment, to raise the quality to the level necessary to render the use of a Big Data system possible, permissible and desirable.

References

[1] http://www.forbes.com/sites/adrianbridgwater/2015/01/19/how-to-test-drive-big-data-analytics/#5ac4b19d728e

[2] https://www.bitkom.org/Publikationen/2016/Studien/Big-Data-Studie/Bitkom-Research-KPMG-Mit-Daten-Werte-schaffen-10-06-2016-final-2.pdf

[3] https://en.wikipedia.org/wiki/Verification_and_validation

[4] https://en.wikipedia.org/wiki/Big_data

[5] http://www.midp.info/uploads/1/0/6/5/10650753/position_paper_-_final.pdf

[6] http://www.informationweek.com/big-data/big-data-analytics/8-reasons-big-data-projects-fail/a/d-id/1297842

[7] http://go.sap.com/docs/download/2015/09/541ccd61-437c-0010-82c7-eda71af511fa.pdf

[8] http://www.softwaretestingstandard.org/

[9] http://www.softwaretestingmagazine.com/knowledge/big-data-how-to-test-the-elephant/

SQS Software Quality Systems AG
Phone: +49 2203 9154-0
Fax: +49 2203 9154-55
info@sqs.com | www.sqs.com