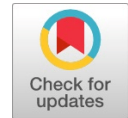


Towards an Automated Testing Framework for Big Data

Ramanathan Meyappan, Nachiyappan S, Disha Nair



Abstract— *Big data testing services are to deliver end to end testing methodologies which address our big data challenges. The testing module includes two types of functionalities. One is functional testing and second is non-functional testing. The functional testing should be accomplished at every stage of big data processing. Functional testing is nothing but the big data sources extraction testing, data migration testing and big data ecosystem. Testing which completes ETL test strategy, Map job reduce validation, multicore Data integration validation and data duplication check. On the other side the non-functional testing is to ensure that there are no quality defeat in data and no performance related issues. It covers the area for security testing, performance testing which solve the problem of monitoring and identify bottlenecks.*

Keywords: *Big Data, Functional Testing, Non-Functional Testing, Data duplication check.*

I. INTRODUCTION

The term “Big data” defines the huge volume of data both structured and non-structured that in updates a business on a daily basis. But it’s not the amount of data that’s important but it’s what the organization with the data matters.

It can be analyzed for insights that lead to better decisions and strategic business moves.

Today, Most of the organizations are collecting, storing, and analyzing massive amount of data refer to as a big data because velocity from which it arrives. Big data comprises of 5 V’s: Volume, Variety, Velocity and Value. More of it comes quickly and more it forms. Volume is the enormity of data, variety is defined as heterogeneity of data, velocity is known as the rate of transfer of data that comes in and goes out, and veracity is the verisimilitude of the data or information .

II. AUTOMATED TESTING

Automation Testing means by using an automation tool to execute some test case suite. The automation software can enter test data into the System Under Test (SUT), compare actual and expected results and generate detailed test reports. It is also known as Test Automation, when tester writes scripts and uses software to test the product. Using a test automation tool it’s possible to record the test suite and re- play as required. Once the test suite is automated, no human work is required. Automating the creation of

both manual test scripts and automated test scripts using a model not only saves effort and thereby cost, but increases coverage and also significantly reduces the time-to-market.

Software vendors must not gamble on the reliability of computer and therefore monitoring cannot be excluded. Automation offers the mechanism to reduce costs and time without impacting performance. Over the past decade, test automation activities have also developed. Although static test scripts gave way to structuring those utilizing sample libraries, information was still embedded and programming expertise was needed to change test cases. New approaches such as user-driven and keyword-driven evaluation source data and even existing system instructions, isolating the software author from scripting difficulties. Such innovations can be effectively applied to reduce the effort and time required for software testing, enabling application providers to be highly successful and offensive. Automation is not directed at eradicating manual supervision, but at reducing the number of manual test cases.

Some of the testing tools used in automation are as follows:

1. Selenium
2. IBM functional tester
3. IBM performance tester
4. Load runner

III. AUTOMATED FRAMEWORKS

In the software testing, automation frameworks are considered to be of significant importance, particularly when you are involved in automation testing². An Automation Framework is gathering of presumptions, thoughts and practices you get while developing the automation project, so it helps in constituting a work stage or support for automated testing. It would be great, if the structure is application independent. In technical terms, an automation framework is a set of strategies, comprises of test tools, hardware, test scripts, methodology, and resources expected to make test automation productive and successful, test results storage, accessing external test resources etc. Automation frameworks can broadly be classified into following.

A. Functional frameworks

Functional testing framework, that is used to test the features or functionality of the system or Software, should cover all the conditions or scenarios including boundary cases and failure paths. It consists of 3 major testing areas namely, white box, black box, and grey box testing where testing without knowledge of internal workings is known as black box testing in which there is no access to source code.

Manuscript published on 30 December 2019.

* Correspondence Author (s)

Ramanathan Meyappan*, TCS, Delaware, USA
ramanathan1.m@tcs.com

Nachiyappan S. SCSE VIT Chennai. nachiyappan.s@vit.ac.in

Disha Nair, Chase Bank, Wilmington USA. Disha.nair1212@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Towards an Automated Testing Framework for Big Data

Tester provides inputs to software and gets output but the coverage area for testing is limited and test cases are difficult to design.

Glass testing or open box testing are ordinarily known as white box testing is a point by point examination of inside rationale and structure of codes however it is exceptionally costly and maintenance is significant bottleneck in this testing. Interestingly, Gray box testing is a test application with restricted hold of the interior workings of an application.

B. Non-Functional Frameworks

The Requirements are tested for its non-functional nature security, performance, user interface are some of the important aspects to be tested.

1. Performance testing is used to determine how fast the system works under a particular workload. It can differentiate two systems to find which executes better. Or it can be measure what part of the system or workload causes the system to perform badly under abnormal conditions.
2. Usability testing, basically it tests the ease with which the user interfaces can be used. Testers tests that whether the application or the product built is user-friendly or not.
3. Security testing is to check that whether the application is secured or not. Non Functional testing is a process which is used to find that a system having information protects the data and maintains the functionality as intended.

C. Data Driven

Data Driven is the approach where the variables are utilized to hold the test information. At runtime, these factors could be stacked from an outer information source like CSV records, legacy data pools, tweaked test information made by running scripts etc¹¹. This approach diminishes the issue of hard coding in test scripts. Take note of that ID of GUI components is still hard coded; for example, the script may contain guidelines that successfully mean.

D. Keyword driven

In this approach, the input, client activities and expected output are encoded utilizing keywords that are normally free of the Application under test (AUT). A test case is encoded as a record made out of these keywords. Test suites made out of such test cases are normally stored in tables. As a feature of the framework development, scripts are composed to make an interpretation of these records to a particular AUT. This approach reduces the problem of hard coding and also provides modularization .

E. Hybrid

This approach consolidates the two methodologies laid out above, and acquires benefits got from both. Over a time frame, cross breed structures have risen as the true standard for computerization necessities. In view of our experience, we prescribe genuine assessment of the hybrid approach amid system outline.

The requirement for a framework automation test suite can be worked by essentially recording different test cases. In any case, it is frequently conceivable to fundamentally

improve the reusability and maintainability of such suites, by creating systems for automation .

F. Reusability

Consider an AUT that has a combo with five conceivable qualities. Facilitate; expect there are five separate test cases, each utilizing alternate values, however indistinguishable in different regards. On the off chance that we utilize the 'record and replay' approach, we would need to record five distinctive test cases. Rather, we can extract the combo as a argument to be passed to the script, and after that call a similar script five times with various arguments.

There are certain user actions – log on, for instance – that might be common to several test cases. Again, these actions (sequences) can be abstracted out and reused in several automated test cases, rather than recording the same sequence multiple times. Short action sequences can also be used to compose long ones

G. Maintainability

Consider a site (the AUT) that has an arrangement of normally required connections showing up in a few web pages. Further assume that we have automated the testing of this website. In the absence of any frameworks, the identification of the links will be arbitrary: it could be indexed on the current page layout, for instance. In a future rendition of the site, these lists could change – say, because of the expansion of different connections. For this situation, every test script including a website page that has this arrangement of connections will be broken, and will require amendment. This could mean a huge revise exertion, as there could be numerous such pages and maybe the entire site. Rather, if named references are given to these connections and just these names are alluded by the test scripts; then changes will be required just to (re)map the names to the genuine connections. Conceptually, a framework eliminates 'hard coding' and provides 'modularization'. Based on our experience, we estimate that in the long run, up to 50% of the script development and maintenance effort can be saved by investing in creating an automation framework.

IV. BIG DATA TESTING

A. Overview of Big Data

Huge information is a relative term depicting a situation where the volume, variety and information or data exceed organizations storage or process constrain concerning exact and timely decision making. Some of this information is held in transactional information stores – the impact of quickly developing on the web movement. Machine-to-machine associations or interactions, such as metering, call detail records, environmental sensing and RFID systems; generate their own tidal waves of data. Every one of these types of information is extending, and that is combined with quickly developing surges of unstructured and semi-organized information from web-based social media¹. The way toward analyzing huge information sets to reveal obscure relationships, hidden patterns, market trends, customer preferences and other valuable data about the business.

The investigative discoveries can prompt to better client benefit, more compelling promoting, enhanced operational effectiveness, new income openings and upper hands over adversary associations and different business benefits.

To analyze data with large volume, big data analytics is usually performed by specialized software tools and applications for data mining, predictive analytics, text mining, and data optimization. Collectively these processes are separate but highly un-segregated functions of high-accomplished analytics. Using BD software and tools enables an organization to process very large volumes of data that a business has collected to determine which data is useful or relevant and can be used analyze and to drive better business decisions in the future.

Tools used in BD scenarios

1. **NoSQL:** Couch DB, mango DB
2. **Map-reduce:** Hadoop, Hive, pig
3. **Storage:** S3, HDFS
4. **Servers:** Google app engine
5. **Processing:** R, Big sheets

BD testing is actually the verification of its data processing. QA personnels verify the successful processing to petabyte of data using commodity cluster and other supportive components. Simply, we can divide the BD testing in 3 steps:

1. Firstly, relevant and correct data is pulled into the system
2. Then, measure up to source data with the data with the data landed on Hadoop or any other platform for data processing.
3. At last, check the data which is extracted and loaded into the proper location in the file system.

The challenge in handling a variety of data is mitigated by the infrastructure, upon which the data are being stored in Big Data implementation. For example, Apache Hadoop uses the HDFS (Hadoop Distributed File System), a dependable shared storage system which can be analyzed using Map Reduce technology.

For big data testing methodology to be successful, the “4Vs” of big data — volume (size of information), variety (distinctive types of information), velocity (investigation of gushing information in microseconds) and veracity (sureness of information) — must be ceaselessly checked and approved. With huge volumes of heterogeneous and unstructured big data increases the complexity of validation, rendering testing based conventional QA system infeasible. Setting up a QA framework to deal with these volumes itself is a test. The nonappearance of vigorous test information administration methodologies and an absence of execution testing apparatuses inside numerous IT companies make enormous information testing a standout amongst the most baffling specialized recommendations that business experiences. Meeting the big data testing challenge requires utilities and computerization answers for enhance test scope, especially when inspecting based conventional QA procedures are deficient

B. Traditional VS Big Data Testing

Big data is a collection of substantial datasets that can't be handled utilizing legacy computing techniques. Testing of these datasets includes different tools, frameworks and techniques to process. Enormous information identifies with data creation, stockpiling, recovery and analysis that is remarkable in terms of volume, variety, and velocity. Testing this will be entirely different from the traditional testing.

Traditional testing and big data has three common properties such as data, infrastructure and validation tools. Testers work with structured data like RDBMS in traditional testing, here big data testers works with both structured and unstructured data like XML and log files. It does not require special test environment to test for ordinary applications where big data testing needs a specialized test environment though the size of data is large. Testers use some UI based functionality or performance test tools to test the system but no such tools are available in market for testing big data.

Table I. Difference based on Some Key Properties

S. No.	Properties	Traditional	Big data
1	Data	Tester is working with structured data	Tester will work with both unstructured as well as structured data
2	Infrastructure	Doesn't require special test environment due to limited size of data	Require special test environment due large size of data
3	Validation tool	Testers used either excel base tools or UI based	No defined tools

C. Layers of Big data

1. Data Source Layer
2. Data Storage Layer
3. Data Processing/Analysis Layer
4. Data Output Layer

1) Data source layer

This stage is the place the information lands at the organization. This can mean the world from email documents, promoting records, online networking channels, feedback, client database, deals records and any information that can sourced from measuring or checking parts of your organization. Keeping in mind the end goal to set up an information procedure, one of the initial steps is taking supply of what you really have and after that measuring it against what you require to answer those inquiries you wish to have helped with. You may discover you have all that you have in any case or it might turn out that you may need to build up new sources of data.

2) Data storage layer

This is the place your Big Data really resides once you have gathered it from your different sources. As the volume of information ventures started to produce and store began to detonate there emerged a need to build up a complex framework that was additionally available.

Devices, for example, Apache Hadoop DFS or Google File System were produced to help with this undertaking. For littler information sets all that might be required is a computer with a big hard disk. However when you begin storing and afterward dissect genuinely big data that your system can comprehend *i.e.* the file system, you will require a framework that sorts out and orders is a way that people can comprehend *i.e.* the database. Hadoop's database is called

HBase however there are various others including MongoDB, Cassandra (utilized by Facebook) and Dynamo DB (utilized by Amazon) all of which depend on NoSQL design. This stage may begin to intrigue the legislature in light of the fact that relying upon what sort of information you are putting away you may well discover there are security and protection controls included.

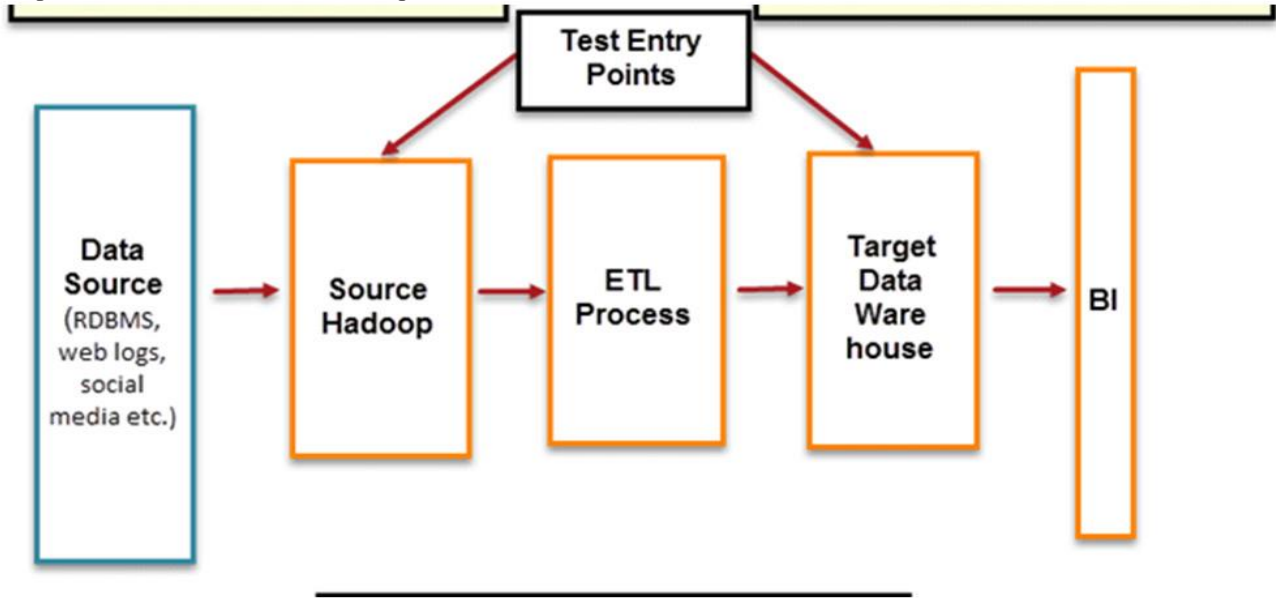


Fig. 1 Testing in Big Data

3) Data processing /Analysis layer

You will need to process and analyze the data if you want to continue to use your stored data to learn something useful. Using the Map Reduce Tool is the most common method. Essentially, what this does is pick the bits of data that you might want to examine and then bring it into a system from which ideas can be gleaned. If your company is big and has the resources to invest in its own data analytics department, it will be part of this level as well. Your group will most definitely use HIVE or Apache PIG to search the information and may use an automatic pattern recognition system to assess patterns and draw conclusions from manual analysis.

4) Data output layer

This is the phase in which analyzes gain insights and are most likely passed on to the people in this organization. It is important to communicate clearly and briefly (especially if the decision maker has no background in statistics). The findings can be in the form of essential suggestions, estimates, graphs and studies. In the final analysis, this is the primary challenge at this stage of the process to demonstrate that tangible improvements can be made in at least one key performance metric.

V. CHALLENGES IN BD TESTING

There are various challenges faced by the big data testers which can be concluded in following factors⁹:

1. Automation test is difficult because of the large range and quantity of data.
2. Due to virtualization, testing again becomes difficult.
3. Again, because of large data sets testing becomes difficult.
4. Proper human resource is required for leveraging Big Data. As in data analysis specialists are required who are

good in business understanding and are capable of dealing with large quantity of data.

5. Too much stress on technical aspects of Big Data than on analytics.
6. Lack of Technical Expertise and coordination.
7. Various set of technologies: Each sub-part has a place with various innovation and requires testing in segregation
8. Specific tools are not available: A single tool cannot perform the end-to-end testing. For example, for message queues NoSQL might not fit
9. Monitoring Solution: To monitor the entire environment a very limited solutions exists.
10. Diagnostic Solution: Custom solution is required to develop to drill down the performance bottleneck areas

Big data services are to deliver end to end testing mythologies which address all our data challenges. Functional testing should be accomplished at every stage of big data processing.

1. BD sources extraction testing is the data processing or the ETL test strategy with data extraction validation included with map reduce job validation.
2. Strategy and source to target field validation comes under data migration testing with data accuracy validation post migration or multicore data integration validation.
3. Big data eco system testing is the metadata and statistical analysis with constraints check and referential integrity with data duplication check.

On the other hand, non-functional testing is to ensure that there is no quality defeat in data and no performance related issues with security test assessment and role based security testing and default permission configuration check.

The testing steps broadly involve the following:

Testing the Input system (HDFS)

To validate if the HDFS has data in the correct format

To validate if all required source data have been moved into HDFS

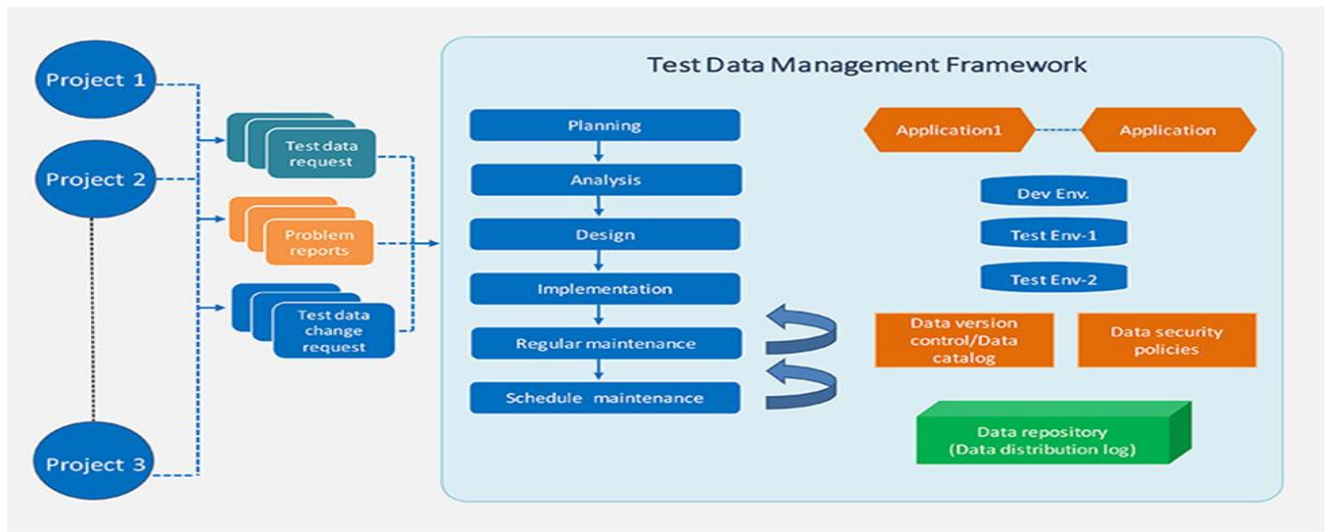


Fig. 2 – Test Data Management Framework

Testing the Output of the MR process

To validate whether the MR process produces the proper output in terms of key value pairs.

To validate if the output is synchronous with the HDFS data source and format Resumed and aggregated data at the output (HDFS / Hive), count to that of the input (HDFS).

If Hadoop is operating with a single node, or on multiple nodes, it is important to have HDFS data aggregation / summarization count with the information moving into the data storage if the HDFS performance is eventually moved to the data warehouse. HDFS can be queried with Pig, while SQL can be used to search the data warehouse

Testing the BI Reports

The template / format & information files are checked here.

ETL Validations

The technique for ETL Validation is the same as for a traditional data store.

VI. BIG DATA TEST ENVIRONMENT

Test system needs to rely on the type of application which you test. For big data experiments, research environments should include sufficient space for storing and processing vast amounts of data, and area of disseminated hubs and knowledge should have a low performance rate with the least CPU and memory.

TDM provide some efficient solution and valuable benefits

TDM implement the organized technology methodology for all possible business situations for testing information specifications or TDM is a method to satisfy test data requirements by ensuring the proper quality test results is delivered in the right standard, correct format and atmosphere at an appropriate time. It can be implemented using well-defined processes, manual and proprietary utilities.

The information entered may be either transactional or static in nature. The main strategy driving TDM is to produce SQL queries that retrieve data from multiple tables and construct flat files with various mapping rules and easily change and merge production data.

It depends upon the superior quality of data with optimal data coverage and detailed analysis and review of synthetic data with less chances of error with clear data security policies. Cost is less with minimum test data storage1.

1. TDM in functional testing covers the positive and negative scenarios with all the boundary conditions and all functional flows. The tool is capable of creating spectrum of data to meet all the data requirements and data can be reused across releases.
2. TDM framework has been tentatively proposed in fig.2.
3. TDM group can provide strategies for the generation of bulk data with refresh cycles with its tools and techniques.
4. The development of test data is instantly attributed to the front end of the user interface (UI) or via software operations that ingest data too quickly, generated or modified in the container.

VII. DATA WAREHOUSE TESTING

The proposed architecture of the data warehouse testing is shown in the fig. 3 which depicts the some entry points in the data warehouse at different layers or we can see that some layers for example data source layer, data extraction layer. ETL layer and data storage layer for which there is a unit test, integration test, regression test and performance test for each layer separately.



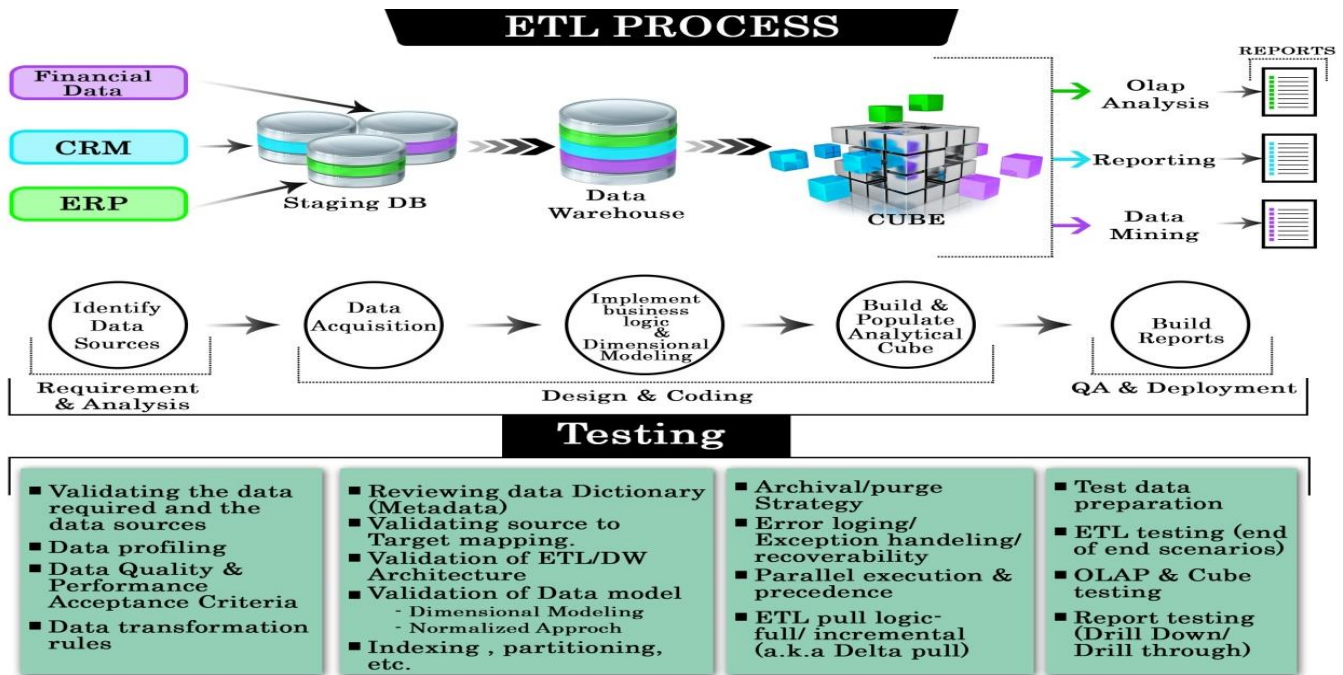


Fig. 3. Data Ware House Test entry Points

DWH projects can be considered as a sequence of data transformation, changes and collection of an arrangement of process. Be that as it may, this straightforward chain of information development prompts to intricacies in testing. For each change of a dataset, testing must guarantee that the change is ideal by including the change rationale into test scripts. With no front end screens, most test scripts must be made as backend scripts (say SQL inquiries) for testing. Along these lines, DWH testing is more escalated and more automatic than normal application testing and requires broad space learning and DWH ideas to make test scripts. There is no promptly accessible UI to visually assess and approve.

A typical DWH implementation will have three core modules, namely:

1. ETL (Extraction, Transformation and Loading).
2. Data Warehouse.
3. Reporting and Analysis packs.

These three modules are interlinked with the organization networks and it can use multiple technology products from multiple vendors to make up a single implementation with the skills of bigdata experts, the pain can be reduced and the learning curve accelerated in three important ways:

1. QA development strategy for battling 4Vs end-to-end.
2. Guidance on the use of proven and appropriate resources.
3. Mitigating risk and related issues suddenly.

VIII. CONCLUSION

Test automation, which is planned and utilized successfully, can be an effective apparatus in an organization's inventory to help deliver quality software faster at a small amount of cost. In any case, at last, take note of that test automation ought to just be considered as an uncommon arrangement of programming that attempts to check the condition of another bit of programming. This paper conveys the features of automated testing techniques and frameworks. It also analyzes how to work with different frameworks for testing of big data. It consists of some basic outlines about big data

and big data testing strategies followed by some points regarding data warehouse testing. Our future work is to design a framework for testing the Big data and moreover, moving directly to the big data testing the initial stage is to first go to the data warehouse testing, work on the different layers of data warehouse by working on small test cases for each layer in a different scenarios including test data for each layer. By analyzing the results of data warehouse testing the big data testing framework will be proposed.

REFERENCES

1. S.Nachiyappan, Dr.S.Justus. Getting ready for Big Data Testing: A practitioner perception. IEEE 4th ICCNT 2013, July 4-6-2013, Tiruchengode, India.
2. Muthuraman Thangaraj and Subramanian Anuradha. State of art in testing big data. International Conference on Computational Intelligence and Computing Research, 2015.
3. Harry M. Sneed and Katalin Erdoes. Testing the Big Data. IEEE Eighth International conference on Software Testing, Verification and Validation Workshops (ICSTW) 13th User Symposium on Software Quality, Test and Innovation. 2015 (ASQT 2015) 978-1-4799-1885-0/15/\$31.00 © 2015 IEEE.
4. Shaik Mohammad Shahabuddin, Y. Prasanth. Integration Testing Prior to Unit Testing: A Paradigm Shift in Object Oriented Software Testing of Agile Software Engineering. Indian Journal of Science and Technology. 2016.
5. Piyaporn Samsuwan and Yachai Limpiyakorn. Generation of Data Warehouse Design Test cases. IT convergence and security 5th international conference on 24-27 Aug, 2015.
6. Sushil Kumar Singh, Sudeep Tanwar. Analysis of Software Testing Techniques: Theory to Practical Approach. Indian Journal of Science and Technology. 2016
7. Data Warehouse Testing Solutions. [https://www.infosys.com/IT-services/validation-solutions/Document s/data-warehouse-testing-solution.pdf](https://www.infosys.com/IT-services/validation-solutions/Document%2Fs/data-warehouse-testing-solution.pdf). Date Accessed 11/11/16.

8. Big Data Testing Services. <https://www.infosys.com/IT-services/validation-solutions/service-offerings/Documents/big-data-testing-services.pdf>. Date Accessed 27/10/16.
9. Proven testing techniques in large data warehousing projects. http://www.syntelinc.com/sites/default/files/syntel_testing_capability_inputs_bidw.pdf. Date Accessed 1/11/16.
10. Test data management in software testing life cycle. <https://www.infosys.com/it-services/validation-solutions/white-papers/documents/test-data-management-software.pdf>. Date Accessed 5/11/16.
11. Big Data Testing Functional and Performance. <http://www.guru99.com/big-data-testing-functional-performance.html>. Date Accessed 7/11/16.
12. Test Automation “to be or not to be”. <http://www.agreeya.com/WhitePaper/TestAutomation.pdf>. Date Accessed 5/11/16. Functional test Automation. http://geometricglobal.com/wp-content/uploads/2013/03/Geometric_Functional_Test_Automation_White_Paper_March_20093.pdf. Date Accessed 5/11/16.
13. Building a Robust Big Data QA Ecosystem to Mitigate Data Integrity Challenges. <https://www.cognizant.com/InsightsWhitepapers/building-a-robust-big-data-qa-ecosystem-to-mitigate-data-integrity-challenges-codex907.pdf>. Date Accessed 5/11/16.
14. The Emerging Big Data System - Testing Perspective. <http://hexaware.com/casestudies/da-it-wp-3.pdf>. Date Accessed 18/11/16.
15. The Big Data Blog. <http://xtreamit.com/the-4-key-layers-of-big-data>. Date Accessed 21/11/16.

AUTHORS PROFILE



M. Ramanathan is working as a Deputy general Manager in TCS for CITI bank Client in Delaware. He has Rich experience in developing IT Projects for more than 20 Years. He has his master’s degree in Bharathiyar University.



Prof. S. Nachiyappan is working in VIT University Chennai campus, Completed his PG in Anna university in 2004 and his area of research is software engineering and Big Data. He is having 5 years of Industry Experience and 10 + Years of teaching experience. He is a member of ACM professional Chapter.



Disha Nair is working for Chase Bank Wilmington, USA. She has rich experience in developing and testing the IT projects over 12 years.