International Journal of Economic Research

ISSN : 0972-9380

available at http: www.serialsjournals.com

Volume 14 • Number 16 (Part 3) • 2017

Towards An Automated Testing Framework for Big Data

Drishti[a] S. Nachiyappan[a] and Justus Selwyn[a]

[a]*School of Computer Science and Engineering, VIT University, Chennai, TN, India,*
E-mail: drishti.2016@vitstudent.ac.in, nachiyappan.s@vit.ac.in, justus.s@vit.ac.in

Abstract: Big data testing services are to deliver end to end testing methodologies which address our big data challenges.

Methods: 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.

Applications: 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.*

1. INTRODUCTION

The term "Big data" defines the huge volume of data both structured and non-structured that inundates 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[1].

2. 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[6].

It is imperative that software vendors do not compromise on software quality, and therefore testing cannot be avoided. Automation provides the lever to cut cost and time without compromising on quality. Test automation practices, too, have evolved over the last decade. While linear test scripts gave way to structured ones that use test libraries, data was still embedded and modifying test cases required programming knowledge. Newer approaches like data-driven and keyword-driven testing source the data and even the directives from external files, insulating the test designer from the complexities of scripting[10].

These advancements can be powerfully combined to reduce the effort and time required for software testing, allowing software vendors to be very competitive and aggressive. Goal of Automation is not eliminate manual testing all together but to decrease number of test cases to be run manually and

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

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

3. AUTOMATED FRAMEWORKS

In the software testing, automation frameworks are considered to be of significant importance, particularly when you are involved in automation testing[2]. 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.

3.1. Functional and Non-Functional Frameworks

3.1.1. 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. 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.

3.1.2. 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.

3.2. 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 datapools, 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.

3.3. 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[11].

3.4. 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[11].

3.5. 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[11].

3.6. 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[11].

4. BIG DATA AND TESTING

4.1. Overveiw 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 media[1].

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[1].

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[9]. 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[9].

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[1]. 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.

4.2. Difference between tradiditional and big data testing

Big data is an 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[9].

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 1
Difference based on Some Key Properties

S. No.	*Properties*	*Traditional DB testing vs. BD testing*	
		Traditional	*Big data*
1.	Data	Tester is working with structured data	Tester will work with both un structured as well as structured data
2.	Infrasture	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

4.3. Layers of Big data

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

4.3.1. 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.[14].

4.3.2. 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[14].

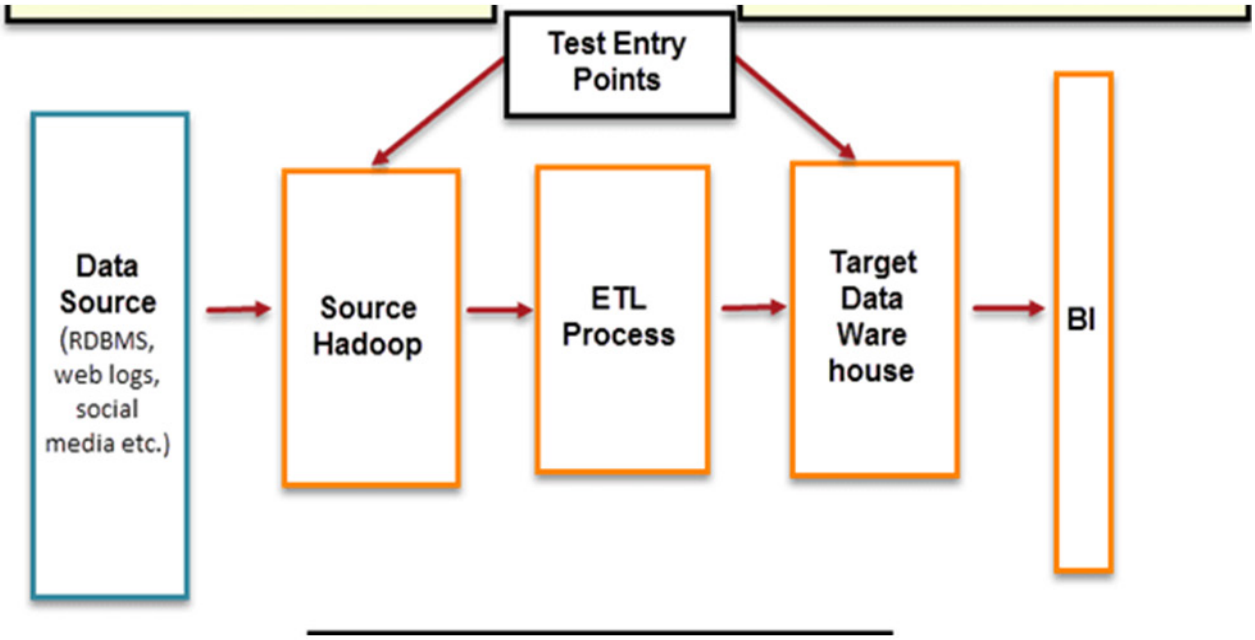

Figure 1: Testing in BD

4.3.3. Data processing /Analysis layer

When you want to begin making use of your stored data in order to understand something useful, you will need to process the data and analyze it. The most common method is through using the Map Reduce Tool. What this does is basically select the bits of data you may want to analyze and then puts it into a format from which insights can be gleaned. If your organization is large and has the resources to invest in its own data analytics team, then they too will also be part of this layer. Your team will most likely make use of HIVE or Apache PIG to query the data and perhaps use an automated pattern recognition tool that has the ability to establish trends as well as draw conclusions from their own manual analysis[14].

4.3.4. Data output layer

This is the stage where insights are gleaned from analysis and then passed on to the people in the organization most likely to benefit. It is important to communicate in a clear and concise manner is critical (especially if the decision maker doesn't have a statistics background). The output can take the form of key recommendations, figures, charts and reports. This is ultimately the main task at this stage of the process is to show how measurable improvement can be achieved in at least one key performance indicator by taking action on the analysis that has been carried out[14].

4.4. Challenges in BD testing

There are various challenges faced by the big data testers which can be concluded in following factors[9]:

1. Automation test is difficult because of the large range and quantity of data.
2. Due to virtualization, testing again become difficult.
3. Again, because of large data sets testing become 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[9].

4.5. Big data testing services

Big data services are to deliver end to end testing methologics which address all our data challenges. Functional testing should be accomplished at every stage of big data processing[5].

1. BD sources extraction testing is the data processing or the ETL test stratergy with data extraction validation included with map reduce job validation[6].
2. Strategy and source to target field validation comes under data migration testing with data accuracy validation post migration or multicore data integration validation[4].
3. Big data ecosysytem 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 are 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[13]:

1. Testing the Input system (HDFS)
    a) To validate if the HDFS has data in the correct format
    b) To validate if all required source data have been moved into HDFS
2. Testing the Output of the MR process
    a) To validate if the MR process generates the correct output in terms of Key-value pairs
    b) To validate if the output is in sync with the HDFS source in data and format
    c) The summarized and aggregated data in the output (HDFS/Hive), tally with that of the input (HDFS)
    d) To validate if specific data transformations are as per the requirement
    e) To make sure that the output remains the same if Hadoop is run on a single-node or on multiple nodes
    f) When the HDFS output is finally moved into the data warehouse, it is required that HDFS data aggregation/summarization tally with the data moved into the data warehouse. HDFS can be queried using Pig, while the data warehouse can be queried using SQL
3. Testing the BI Reports
    a) Here we validate the reports for layout/ format and data
4. ETL Validations

The ETL Validation strategy remains the same as that of a typical data warehouse[13].

5. BIG DATA TEST ENVIRONMENT

Test Environment needs rely on upon the kind of application you are testing. For Big data testing, test environment ought to include enough space for storage and process vast measure of information, environment ought to have group with disseminated hubs and information and it ought to have least CPU and memory use to keep performance high[9].

5.1. TDM provide some efficient solution and valuable benefits

TDM introduces the structured engineering approach to test data requirements of all possible business scenarios or TDM is a process of fulfilling the test data needs of testing team by ensuring that the test data of the right quality is provisioned in suitable quality, correct format and proper environment at the appropriate time. It can be implemented with the aid of well- defined processes, proprietary utilities and manual methods.

Input data can be transactional or static in nature. Basic strategy behind the TDM is creation of SQL queries that extract data from multiple tables in the database and creation of flat files with different mapping rules and simple modification of production data and combination of all these[1].

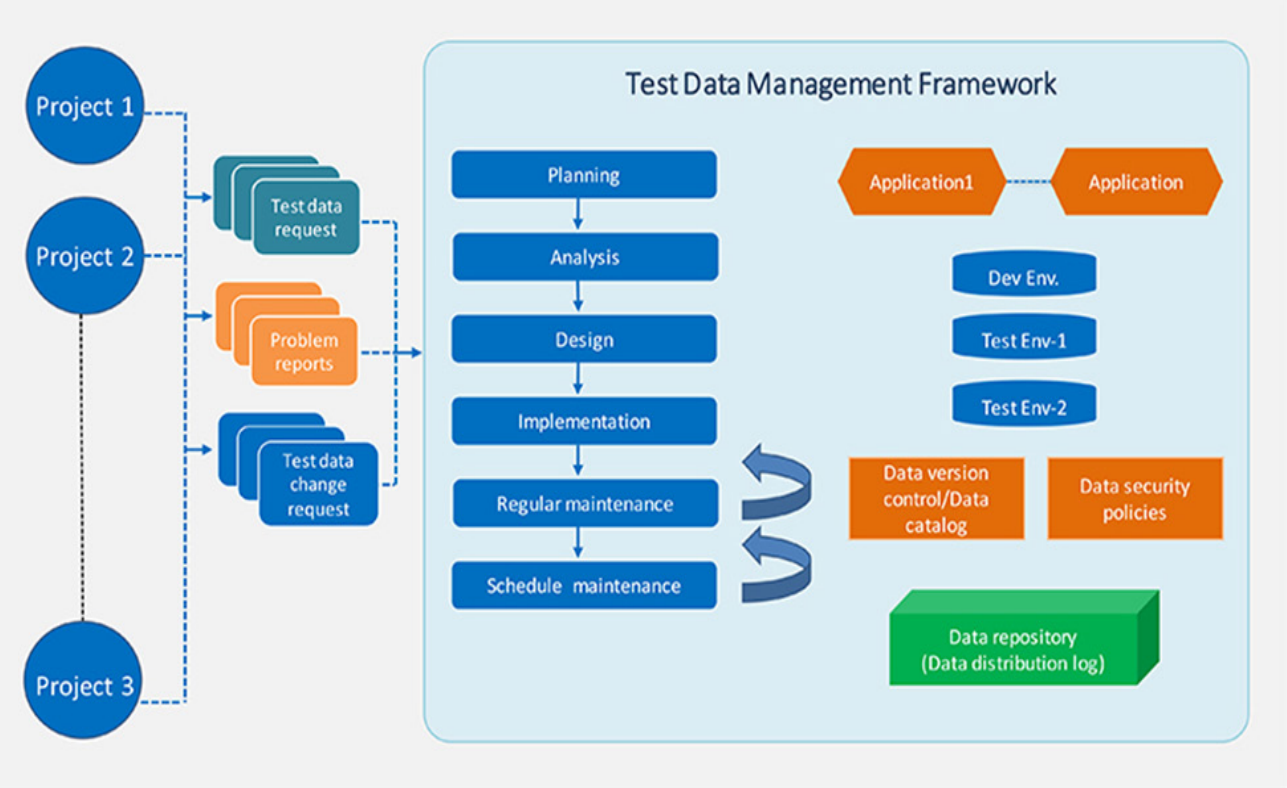

Figure 2:Test data management framework

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 storage[1].

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 team with their tools and techniques can provide solution for bulk data generation with refresh cycles.
4. Automation of test data creation is due to user interface (UI) front end or via creates or edits data operations in the database which consume data so rapidly[1].

5.2. 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[5].

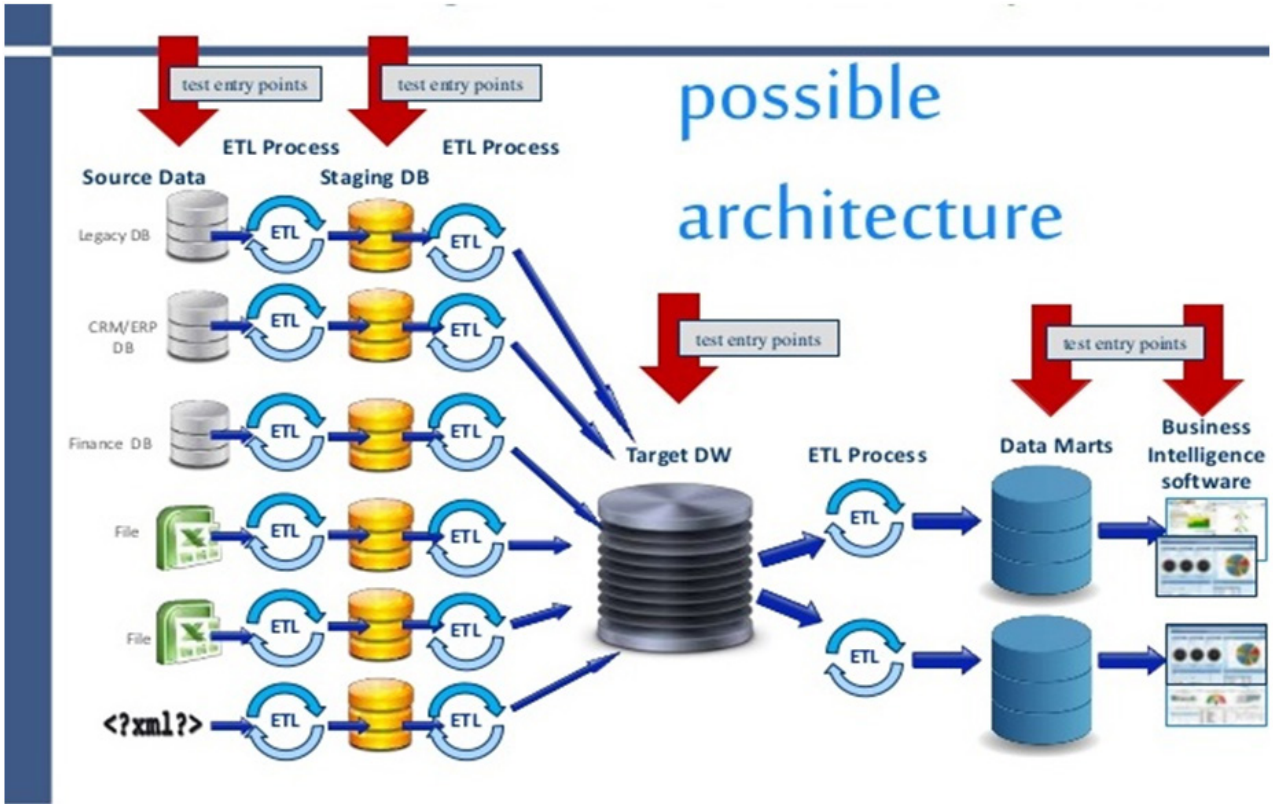

Figure 3: Dataware house test entry points

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[5]. Utilizing the skill of big data testing specialists can ease the pain and accelerate the learning curve in three important ways:

1. End-to-End QA design technique to battle with the 4Vs.
2. Gaining up guidance on the utilization of proven and appropriate tools.
3. Mitigating unanticipated (and anticipated) risks and related issues[12].

6. 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 ICCCNT 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/Documents/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] Teat 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.

[13] 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.

[14] 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.

[15] The Emerging Big Data System - Testing Perspective. http://hexaware.com/casestudies/da-it-wp-3.pdf. Date Accessed 18/11/16.

[16] The Big Data Blog. http://xtreamit.com/the-4-key-layers-of-big-data. Date Accessed 21/11/16.