International Journal of Information Technology and Knowledge Management
July-December 2012, Volume 5, No. 2, pp. 239-243

A CASE STUDY ON REGRESSION TEST AUTOMATION FOR DATA WAREHOUSE QUALITY ASSURANCE

Manjunath T.N.[1], Ravindra S. Hegadi[2], Yogish H.K.[3], Archana R.A.[4] and Umesh I.M.[5]

ABSTRACT: In current trend, every software development, enhancement, or maintenance project includes some quality assurance activities. Quality assurance attempts defects prevention by concentrating on the process of producing the rather than working on the defect detection after the product is built. Regression testing means rerunning test cases from existing test suites to build confidence that software changes have no unintended side-effects. Data warehouse obtains the data from a number of operational data source systems which can be relational tables or ERP package, etc. The data from these sources are converted and loaded into data warehouse in suitable form, this process is called Extraction, Transformation and Loading (ETL). In addition to the target database, there will be another data base to store the metadata, called the metadata repository. This data base contains data about data-description of source data, target data and how the source data has been transformed into target data. In data warehouse migration or enhancement projects, data quality checking process includes ensuring all expected data is loaded, data is transformed correctly according to design specifications, comparing record counts between source data loaded to the warehouse and rejected records, validating correct processing of ETL-generated fields such as surrogate keys. The quality check process also involves validating the data types in the warehouse are as specified in the design and/or the data model. In our work, have automated regression testing for ETL activities, which will saves effort and resource while being more accurate and less prone to any issues. Author experimented around 338 Regression test cases, manual testing is taking around 800 hrs so with RTA it will take around 88 hrs which is a reduction of 84%. This paper explains the process of automating the regression suite for data quality testing in data warehouse systems.

Keywords: ETL, Mapping, Regression Testing, DWH, RTA

1. INTRODUCTION

The main function of an ETL tool is to extract the data from different sources, transform the data and loading into data warehouse. An ETL scripts used as automation tool for Data Quality and Data consistency testing. Important advantages of this are: (*i*) It reduces testing efforts (*ii*) It can be re-usable logic for one or more test cases (*iii*) It saves testing time, more accurate and etc. In conventional Data Quality testing, to test multiple source systems against single target system we need to extract the data from multiple source systems and save it in spread sheet[6][7]. Then we need to compare this data against target system data. This consumes more time and testing accuracy is less. But with an ETL Scripts (Mappings) we can automate this, this saves the time and it's more accurate than manual testing. And also it's easy to operate. There are distinct points in ETL process where data quality can be injected, the first is when data is extracted and the other is when the data is cleaned and conformed. In the data model, TEST_CASE table contains all the test cases which need to be automated and assigned unique id for each test case called TEST_CASE_ID. TEST_RESULT table contains result of each test case run i.e. PASS or FAIL, TEST_RUN_ID it's a unique identifier for each test case run, RUN_DATE date and time of the test case run, ERROR_MESSAGE it's a message that contains why test case is failed for passed test case value is null. Last table is REPORT table; it contains delta records (un-matched records) between Actual and Test data. Structure of this table is same as target table along with the new column TEST_RUN_ID. Using Informatica tool author created mapping for each test case i.e. one mapping for one test case (for some test cases, we have combined more than one test case into one mapping e.g. column wise test cases for same table). These mappings are validating test data versus actual data, if both are matching then it's inserting a record into TEST_RESULT table with new TEST_RUN_ID, TEST_CASE_ID of the test case and RESULT = PASS. If both are not matching then its inserting a record into TEST_RESULT table with new TEST_RUN_ID, TEST_CASE_ID of the test case and RESULT = FAIL and also it's going to capture DELTA records (un-matching records) into DELTA table.

2. LITERATURE REVIEW

In the paper titled 'Data Quality Tools for Data Warehousing-A Small Sample Survey Using Information

[1,3,4] Research Scholar, Bharathiar University, Coimbatore, Tamil Nadu, India,
[1]*E-mail: manjunath.tnarayanappa@rediffmail.com,* [3]*yogishhk@gmail.com,* [4]*archana.ra@rediffmail.com*

[2] Department of Computer Science, School of Computational Science, Solapur, University, India,
E-mail: ravindrahegadi@rediffmail.com

[5] Department of Information Science, RVCE, Bangalore, Karnataka, India, *E-mail: umesh.mphil@gmail.com*

in Government Program' [1] by Center for Technology in Government University at Albany / SUNY, there is a estimation that as high as 75% of the effort spent on building a data warehouse can be attributed to back-end issues, such as readying the data and transporting it into the data warehouse (Atre, 1998). Data quality tools are used in data warehousing to ready the data and ensure that clean data populates the warehouse, thus enhancing usability of the warehouse. This research focuses on the problems in the data that are addressed by data quality tools. Specific questions of the data can elicit information that will determine which features of the data quality tools are appropriate in which circumstances. The objective of the effort is to develop a tool to support the identification of data quality issues and the selection of tools for addressing those issues. A secondary objective is to provide information on specific tools regarding price, platform, and unique features of the tool .In the article titled 'Money Saving Tip: Use ETL Tool for Data Quality'[2], Loraine Lawson states that one can use your ETL (extract, transform and load) tool for detecting data problems, cleansing the data and even maintaining data quality. He states ETL may not be good enough-more on that later-or perhaps the organization has invested in a full data quality solution. Sometimes it works, sometimes it doesn't. The Data Quality Pro piece warns it isn't a replacement for a high-end data quality solution, but if you want better data, but can't afford a full-blown investment, an ETL can be an excellent starting point for the following reasons: An ETL can address nearly 70 percent of data quality requirements, according to Data Quality Pro, which used Arkady Maydanchik's "Data Quality Assessment" framework as a gauge [8]. That remaining 30 percent, only 7 percent of it is completely non-complaint. It's not perfect, but it's better than 100 percent unsure. You can use the ETL data quality approach to tackle low-hanging fruit problems, and then turn any cost savings into a down payment for a full-blown tool or, at least, that's what the Data Quality Pro piece recommends [10].

3. EXISTING SYSTEM

Any software project will have multiple releases hence more effort is taking for regression testing, for each release the regression testing is taking around 800 hrs effort to complete around 338 test cases [8]. Manually creating the test script using SQL for each of the test case and manual execution and store the result either in excel or temp DB then compare, which requires lot of time and cost and more manual intervention so more prone to errors[8]. Process diagram is shown in figure 1.

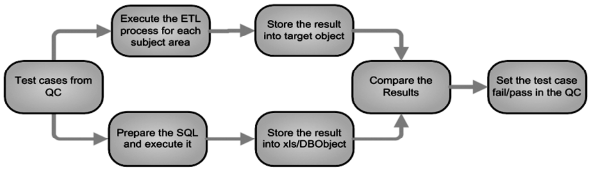
Execute the ETL process for each subject area
Store the result into target object
Test cases from QC
Compare the Results
Set the test case fail/pass in the QC
Prepare the SQL and execute it
Store the result into xls/DBObject

Figure 1: Manual Process of Testing

4. PROPOSED MODEL

Using the Informatica (ETL tool) and Oracle DB, we have automated the manual process of preparing the SQL, executing and comparing the results, the green colored boxes were automated in reducing the effort and cost.

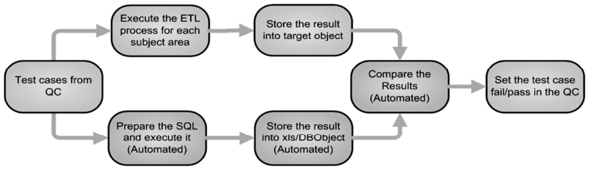
Execute the ETL process for each subject area
Store the result into target object
Test cases from QC
Compare the Results (Automated)
Set the test case fail/pass in the QC
Prepare the SQL and execute it (Automated)
Store the result into xls/DBObject (Automated)

Figure 2: Automation Process of Testing

4.1 Implementation Details

1. Author identified re-usable regression test cases, these test cases can be re-used for any kind of enhancements or any releases.
2. Author created Database tables called RTA_REF_TEST_CASE (This table is used to store all the test cases which need to be automated), RTA_TEST_RUN_RESULT (This table is used to capture result of each Test case run) and RTA_DELTA_* (This table is used to capture the Delta records i.e. un-matching records from source data to target data. * means name of the target table. E.g. if target table is ODS_DIM_TIME then DELTA table name will become RTA_DELTA_ODS_DIM_TIME).
3. Using Informatica we have created mapping for each test case i.e. one mapping for one test case (for some test cases we have combined more than one test case into one mapping e.g. column wise test cases for same table).
4. If Test Case is passed then Informatica mapping is going to insert a record into RTA_TEST_RUN_RESULT table with new TEST_RUN_ID (previous TEST_RUN_ID + 1), TEST_CASE_ID of the test case (from RTA_REF_TEST_CASE) and RESULT = PASS.
5. If test case is failed then Informatica mapping is going to insert a record into TEST_RESULT table with new TEST_RUN_ID, TEST_CASE_ID of the test case, RESULT = FAIL, ERROR_MESSAGE = 'Actual message of the test case' and also it's going to capture

DELTA records (un-matching records between source and target) into DELTA table, this is descripted in figure 3.

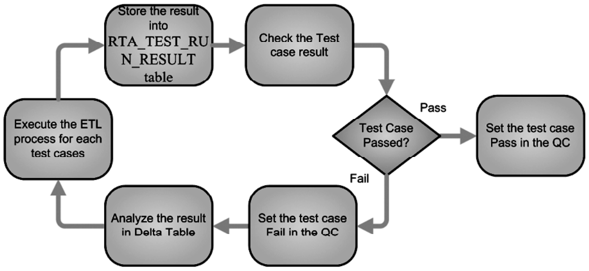
Store the result into RTA_TEST_RUN_RESULT table
Check the Test case result
Test Case Passed?
Pass
Set the test case Pass in the QC
Fail
Set the test case Fail in the QC
Analyze the result in Delta Table
Execute the ETL process for each test cases

Figure 3: RTA Execution Process Flow

5. RESULT AND DISCUSSION

This logic had been tested thoroughly and same had been applied for few re-usable regression test cases. To validate the data quality between file and data base table, manual testing will require more than one day but with this approach it took 0.25 man hours with more accurate result. To check Time Dimension data, manual testing will require nearly one day but with this approach we completed the testing within 0.5 man hours, Using RTA, we executed all independent test cases parallely but this is not possible by manual testing or any conventional method.

Table 1
Test Results Table with Manual and Automation Effort

No of Test cases	*Manual Effort*	*Automation Effort*
10	20	2
20	40	5
30	60	7
40	80	9
50	100	11
60	120	13

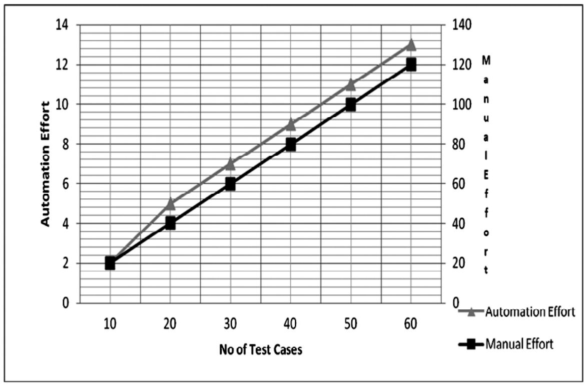

Figure 4: Effort Comparison Between Manual and Automation

5.1 Cutthroat Approaches

With any Test automation tool, it's difficult to test the data quality between source and target system in data warehouse environment but with RTA we can test the data quality between source and target system with more accurate test results.

6. CONCLUSION

Business Requirements will continue to change due to business demands, there is a need to re-test the data when there are some change requests are coming from the customer, so regression test automation is needed to test data warehouse which involves the ETL, Our experiments shows the reduction in test cycle time by 50 to 60 percent which in turn reduces effort and cost by 84 percent, we can integrate RTA with Quality Center, author hopes this study will help Quality Analyst in performing Quality activities for data warehouse system.

REFERENCES

[1] Data Quality Tools for Data Warehousing-A Small Sample Survey Using Information in Government Program by Center for Technology in Government University at Albany / SUNY

[2] Ralph Kimball. The Data Warehouse Toolkit, Wiley India Pvt Ltd., 2006

[3] Dr. K.V.K.K Prasad, Data Warehouse Development Tools, Dreamtech Press, 2006.

[4] Alex Berson, Data Warehousing Data Mining and OLAP.

[5] White Paper by Vivek R. Gupta, Senior Consultant, System Services Corporation, "An Introduction to Data Warehousing".

[6] Manjunath T.N., Ravindra S. Hegadi, Ravikumar G.K. "Analysis of Data Quality Aspects in Data Warehouse Systems". (IJCSIT) *International Journal of Computer Science and Information Technologies*, **2** **(1)**, 2011, 477-485.

[7] Manjunath T.N., Ravindra S. Hegadi, RaviKumar G.K., "Design and Analysis of DWH and BI in Education Domain", *IJCSI International Journal of Computer Science Issues*, **8(2)**, March 2011 ISSN (Online): 1694-0814.545-551.

[8] Manjunath T.N., Ravindra S. Hegadi and Mohan H.S. Article: "Automated Data Validation for Data Migration Security". *International Journal of Computer Applications*, **30(6),** 41-46, September 2011. Published by Foundation of Computer Science, New York.

[9] Article Titled "What Academia Can Gain from Building a Data Warehouse" *by David Wierschem, Jeremy McMillen and Randy McBroom.*

[10] Channah F. Naiman, Aris M. Ouksel (1995). "A Classification of Semantic Conflicts in Heterogeneous Database Systems", *Journal of Organizational Computing*, **5**, 1995

[11] John Hess (1998), "Dealing with Missing Values In the Data Warehouse" *A Report of Stonebridge Technologies*, Inc (1998).

[12] Jaideep Srivastava, Ping-Yao Chen (1999). "Warehouse Creation-A Potential Roadblock to Data Warehousing", *IEEE Transactions on Knowledge and Data Engineering January/February,* 1999, **11(4)**, pp. 118-126.

[13] Amit Rudra and Emilie Yeo (1999). "Key Issues in Achieving Data Quality and Consistency in Data Warehousing among Large Organizations in Australia", Proceedings of the 32nd Hawaii International Conference on System Sciences -1999

[14] Jesus Bisbal et al (1999). "Legacy Information Systems: Issues and Directions", *IEEE Software September/ October* 1999.