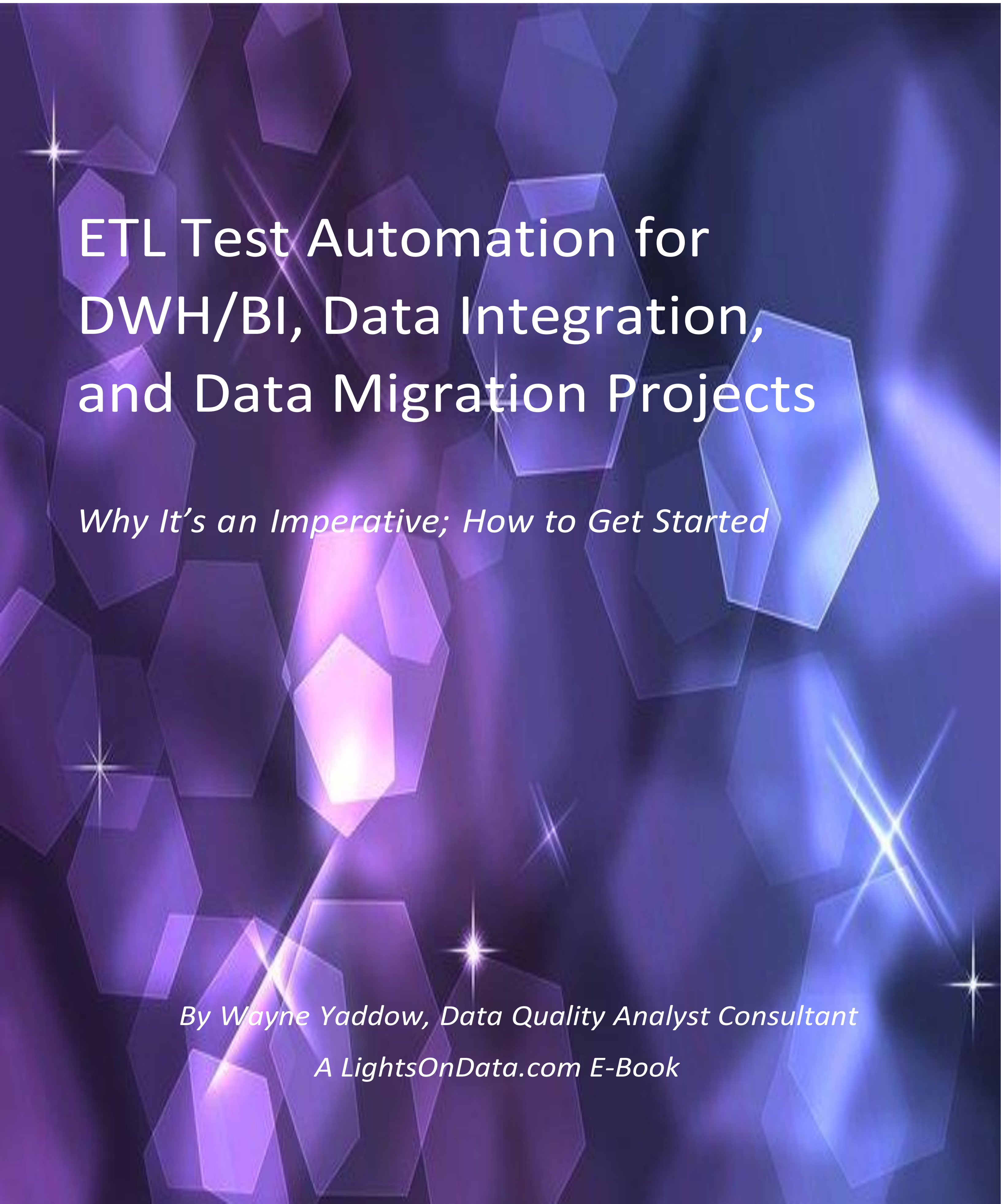

ETL Test Automation for DWH/BI, Data Integration, and Data Migration Projects

Why It’s an Imperative; How to Get Started

By Wayne Yaddow, Data Quality Analyst Consultant

A LightsOnData.com E-Book

Gartner recently stated that between 70 and 80 percent of business intelligence initiatives initially end up failing, but many resume due to their importance to the organization. As businesses create (and need) more data than ever before, the sheer number of BI failures threatens to grow exponentially. This could have a farreaching impact on the underlying digital transformation initiatives that these BI projects are designed to enable.

Given that companies are releasing new applications faster than ever—some releasing updates on demand and multiple times per day—too many organizations are using manual ETL test processes and the wrong tools to manage critical parts of releases for highly visible, often customer-facing applications. That translates into risk: risk to customer loyalty, the brand, confidential data, and critical business decisions.

This paper explores how applying DevOps-style test automation to DWH/BI and other data integration projects can guarantee a high level of data quality—instilling the trust that is essential for the success of BI projects and the digital transformation initiatives that are ultimately driving them.

Taking a DevOps Approach to BI/DWH Testing

DevOps, with its focus on tool automation across the entire development lifecycle, addresses an enormous challenge for "big data" and DWH/BI developers. Many of today's big data and DWH/BI projects are already leveraging (or actively planning to adopt) Agile and DevOps processes—but not so much for testing. DWH/BI projects, in general, are not currently using automated testing tools to the extent that is needed for project successes. Perhaps this is because they believe the required testing functions are not commercially available, or are too complex and expensive to develop in-house.

When thinking about what needs to be tested to ensure data integrity, it's important to consider that BI is more than just data warehouses (DWH) and ETL (Extract Transform Load). Services between the ETL processes, as well as the middleware and dashboard visualizations, also come under the purview of BI. Messages and negotiating pacts between these layers is complex and requires much coordination and testing.

DevOps helps facilitate this with constant deployments and testing. Implementing a DevOps testing approach to DWH/BI means automating the testing of different source and target data sets to keep data current. This can be tremendously beneficial when handling many diverse data sources and volumes—for some projects, hundreds. Your team will be able to detect errors before they threaten BI applications in production. Moreover, you will have more time to fix issues before reaching production.

Why Test Automation?

Continuous quality is a systematic approach to achieving the quality goals of development and the businesses it supports. In the 2018 Magic Quadrant for Software Test Automation, Gartner states: "Test automation tools are essential elements of a DevOps toolchain and enablers for achieving the continuous quality approach required for successful DevOps."

However, as for any IT project, repeated ("regression") testing is important for guaranteeing a high level of (data) quality. The more we test, the more bugs will be resolved before going live. This is especially crucial for business intelligence projects. When the users can't trust the data, it's likely that the BI solution itself will not be trusted…and fail.

As mentioned earlier, ETL testing is primarily conducted manually, which makes it a very labor intensive and error prone process. Automating ETL tests allows frequent smoke and regression testing without much user intervention and supports automated testing on older code after each new database build. Automation can not only help execute tests; it can also assist with designing and managing them.

The decision to implement automated tools for ETL testing depends on a budget that supports additional spending to meet advanced testing requirements. It is important to remember that test tools built and maintained in-house are better than no test automation at all. In the end, test automation will save much time. Additionally, business users will appreciate the quality of BI deliverables and accept the data from the Data Platform solution as the "single version of the truth."

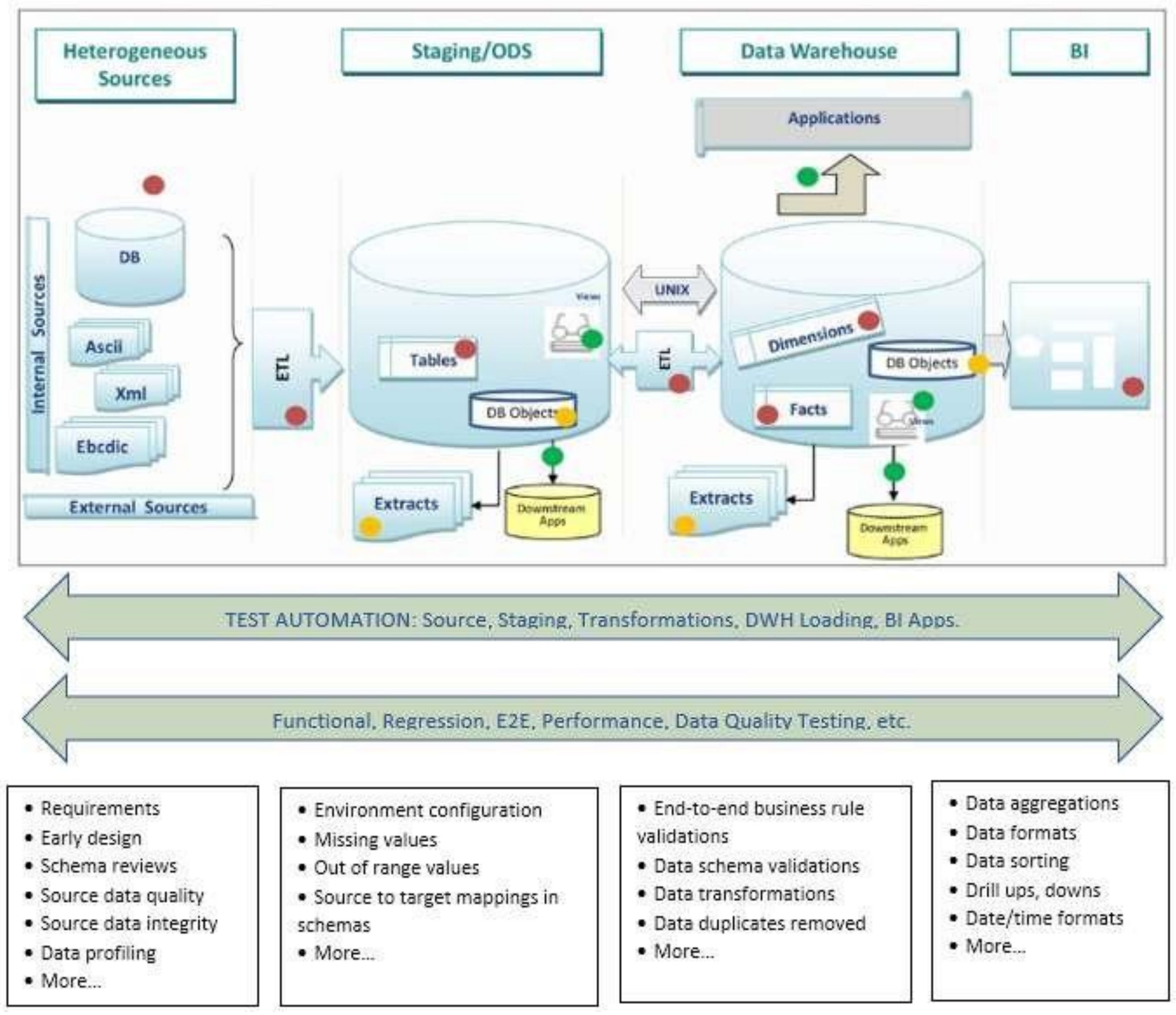

Figure 1: A sampling of tests and validations that should be considered for most DWH/BI projects

Planning for the ETL Test Automation Process

Below are some of the most highly-recommended test automation planning steps for DWH/BI projects. As with all projects, the decisions made during the planning stages of a test automation project set the stage for success—or for failure. For this reason, it’s suggested that you take the necessary time to set goals, analyze current processes, and build the right implementation team prior to launching the test automation project.

1. Analyze your current testing process—from unit testing, to component testing, to data quality testing and beyond.
2. Define the stakeholders and IT team.
3. Identify and prepare several test scenarios for test automation.
4. Research and select two or three top commercial or open source ETL and data quality automation tools for an in-depth evaluation.
5. Conduct proof of concept (PoC) exercises—preferably with the collaboration of tool vendors. Vendors help ensure you’re really understanding the potential of the tool. They can also help you gain the most accurate assessment in the shortest time possible.
6. Implement the selected automation tools.
7. Make time for training and the learning curve.
8. Begin automating already-documented test cases.
9. Plan to review your process and results, then adjust as necessary.

Understanding Which ETL and DWH/BI Verifications are Best for Automation

Regarding test scenarios for test automation, evaluate your scenarios and determine which are the best candidates for automation based on risk and value. Which types of defects would cause you to stop an integration or deployment? Which types of tests exercise critical core functionality? Which tests cover areas of the application that have historically been known to fail? Which tests are providing information that is not already covered by other tests in the pipeline?

Common preferences for manual DWH testing

- Exploratory Testing: This type of testing requires the tester’s knowledge, experience, analytical/logical skills, creativity, and intuition. Human skills are needed to execute the testing process in this scenario.
- Ad-hoc Testing: In this scenario, there is no specific approach. It is often an unplanned method of testing where the understanding and insight of the tester is the important factor.

Common preferences for automated DWH testing

- Source to Target Data Reconciliation Testing (including transformation testing, regression testing, smoke testing): Here, automated testing is suitable because of frequent code changes and the ability to run the regression assessments in a timely manner.
- Repeated Execution: Since this testing requires the repeated execution of a task, it is best automated.
- Load Testing: Load testing is another type of testing where automation is essential for efficiency.
- Performance Testing: Likewise, testing which involves the simulation of thousands of concurrent users requires automation.
- End-to-End Testing: Data testing can be time-consuming because of the variety of stages, technologies and a vast volume of data involved. Each phase of ETL testing requires different strategies and types of testing—for example, one-to-one comparisons, validations of migrated data, validations of transformation rules, reconciliations (e.g., sources to targets), data quality checks, and front end testing of BI reports.

Table 1 below lists most types of tests that are often considered for test automation and test automation tool implementations (commercial, open-source, and in-house tools). Utilizing a list of test scenarios such as this can be a good start on your road to DHW/BI test automation.

Table 1: Test scenarios and test cases frequently considered for automated testing

TEST SCENARIOS	TEST CASES
METADATA VALIDATION	Validate the source and target table structure as per the mapping and metadata documents • Data types are validated in the source and the target systems. • The length of data types in the source and the target system should be the same. • Data field types and their format are verified to be the same in the source and the target system. • Validating the column names in the target system.
VALIDATE MAPPING DOCUMENTS	Validate mapping and metadata documents to ensure all the information has been implemented. The mapping document should have a change log, maintain data types, length, transformation rules, etc.
VALIDATE CONSTRAINTS	Validating all column and transformation constraints and ensuring that they are applied to the expected tables.

DATA CONSISTENCY & INTEGRITY CHECKS	Checking the misuse of integrity constraints like foreign keys – no orphan FK's. The length and data type of an attribute may vary in different tables, although their definition remains the same at the semantic layer.

TEST SCENARIOS	TEST CASES
DATA COMPLETENESS VALIDATION	Verifying that all data is loaded to the target system from the source system. • Record counts in the source and the target data. • Boundary value analysis (tests of min/max, no truncations). • Validating the unique values of primary keys.
DATA CORRECTNESS VALIDATION	Verifying values of data in the target system. • Misspelled or inaccurate numeric data in target table. • Distinct values in columns (not unique data) are stored when you disable integrity constraint at the time of import.
DATA TRANSFORMATIONS APPLIED ACCORDING TO BUSINESS RULES	Creating a matrix of scenarios for input values and expected results and then validating with end users. • Validating parent-child relationships in the data by creating scenarios. • Using data profiling to verify the range of values in each field. • Validating if the data types in the warehouse are the same as mentioned in the data model. • Default values, data trimming, etc. • Verify source table joins for aggregations, etc.
DATA QUALITY VALIDATION	Performing number check, date check, precision check, data check, null checks, etc. on both source and target data. Example - Date format should be the same for all the values per the column definitions
DUPLICATE DATA VALIDATION	• Validating duplicate values in target system columns and rows when data is loaded from multiple columns in sources • Validating primary keys and other columns if there are any duplicate values as per the business requirement. • Verifying that multiple columns specified as a unique key can be grouped without resulting in duplicate records.

TEST SCENARIOS	TEST CASES

DATE VALIDATION CHECKS	Validating the date field for all defined actions performed in ETL processes • From_Dates not greater than To_Dates. • Min and max values within bounds (01/01/1970, 2099-12-31). • Date and time values as specified. • Date values contain no junk values or null values.
CDC, SCD, FACT TABLE UPDATES	Verifying that all changed data (CDC) is captured from sources and applied according to changing dimensions (SCD) and fact table specifications in requirements.
COMPARE STAGING AND DW TABLES	Verifying that target DWH tables are precisely the same as staging where specified and DWH tables are correctly loaded where differences are specified between them.
DROPPED RECORDS	Validating that no records are dropped where they should not be between all sources and targets. Verify that records with error_status = 'E' are dropped and that any records in the same or related tables that with foreign keys to these dropped records are processed according to specifications.
EXTRA RECORDS, ADDITIONAL COLUMNS IN TARGET	Verifying that extraneous data that was not meant to be loaded was not actually loaded.
UNIQUE KEY EXCLUSIVITY	Verifying that all columns specified as a unique key are unique among all records.
RECORD COUNTS	Verifying record counts as correct when compared with source records and when compared from one DB load to another
EXPLORATORY TESTING	Providing distinct values from all columns to support input exploratory testing.
MAINTAIN SOURCE TABLE ID'S THROUGH TO DW	Providing functions that allow tracking / verification of surrogate or native or ID keys from source to the final target.
ETL LOOKUP PROCESSING	Verifying that ETL "lookups" were processed correctly
AGGREGATED VALUES	Verifying aggregation of values from sources to targets

Key Takeaways

- Many projects' DWH/BI teams have found that it's possible to succeed with automated testing.
- Automated testing will not replace all manual unit, component, and end-to-end testing. However, it will ensure that the more costly manual work is focused on high-risk, high-value activities— and, in the process, complement the QA process.
- Creating automated DWH/BI tests is well worth the upfront effort, especially in the data warehouse testing phases. Automated tests can be run hundreds of times at modest cost with almost no physical time constraints.
- We know that testing takes time. We know that testing costs money. If planning and other upfront automation efforts reduce time and costs, that's undeniably beneficial for your organization's bottom line.

About the Author

Wayne Yaddow has 15 years of experience leading data migration/integration/ETL testing projects at organizations including J.P. Morgan Chase, Credit Suisse, Standard and Poor's, and IBM, and Achieve3000. Additionally, Wayne has taught IIST courses on data warehouse, ETL, and data integration testing. He continues to lead ETL testing and coaching projects on a consulting basis. You can contact Wayne at wyaddow@gmail.com.