Suellen Arbuckle

Suellen Arbuckle is a Test Lead with Eli Lilly & Company, in Indianapolis, IN. During her 17-year IT career she was a technical writer for Gould Computer Systems and Motorola in Illinois, and Pythia Corporation and Whittman-Hart of Indianapolis. While a consultant for Whittman-Hart, she worked three years as a technical writer on a variety of projects that included legislative software for the Illinois state legislature, logistics management for Mitsubishi, and several smaller projects. She worked as a consultant on the Y2K program for Eli Lilly & Company on the Communications team. Her contact in that role with the testing services team of Lilly sparked her interest in testing and she became a Lilly Test Analyst in February 2000. One year later she moved into the Test Lead role and has worked on projects involving a Peoplesoft upgrade, automating global marketing plans, and several projects in the discovery and research area of Lilly. It was on a data warehouse project in the discovery and research department that she gained her knowledge of data warehouse testing.

Rebecca Cooper

In 2001, Rebecca Cooper joined the Indianapolis-based Eli Lilly & Company as a test analyst, working on projects involving the automation of global marketing plans and standardization of product team intranet sites. Rebecca temporarily functioned as a data analyst for research and discovery data migration and data warehousing projects prior to her current systems analyst position.

Prior to joining Eli Lilly & Company, Rebecca was employed with USAGroup (now Sallie Mae). During her tenure there, she worked as a mainframe developer using an IDMS database. She also served as the sub-system support lead and development lead on several Department of Education regulatory projects, including electronic transfer applications. In conjunction with these responsibilities, Rebecca served as her team's release coordinator, a position requiring the organization and management of cross-team projects and resources.

Rebecca attended Indiana State University and graduated with Bachelor of Science degrees in both Accounting and Management Information Systems.

How to Thoroughly Test a Data Warehouse

Rebecca Cooper
and
Suellen Arbuckle

Introduction

Purpose

The purpose of this paper is to describe general data warehouse structure and background, as well as specific situations encountered during the testing effort for our project.

Our project was to test a data warehouse and data mart for a large research department of a pharmaceutical manufacturing company. This was a pilot project and our efforts, if successful, would pave the way for future data warehouse projects within the company.

Our role on the project team was to provide quality through validation of the business rules. We met this challenge by determining if the extract/transform/load (ETL) process was functioning in the approved manner. We created basic principles that were used as a starting block for our process.

Data Warehousing

Data warehousing is the process of managing a data warehouse and any data marts associated with a specific enterprise (business) layer. It was important throughout the warehousing process to maintain focus on the 'big picture' – the integrated data model of the enterprise.

The data warehousing environment does not have to be complex. It can be as simple as two levels: the enterprise level and the warehouse level. Adding additional levels to the previous example such as a data mart, can increase the complexity of the environment.

Data Warehouses and Data Marts

A data warehouse is a cohesive data model that defines the central data repository for an organization.

A data mart is a data repository for a specific user group. It contains summarized data that the user group can easily understand, process, and apply. A data mart cannot stand alone; it requires a data warehouse.

Because each data warehousing effort is unique, your company's data warehousing environment may differ slightly from what we are about to introduce.

Project

Project Makeup

The following list describes the makeup of our project and some critical questions we needed to answer.

Where	How far will this warehouse reach? Are there gaps within the enterprise level that we should be incorporating in our warehouse? To the best of our knowledge, have we identified all the possible impacts to the local environment? * Adjoining environment * Applications
What	Do we understand the vision of the project and how it aligns with the project or department mission?
Why	Are we maintaining data quality expectations to support the business goals?
Who	How many resources should make up the project team? At a minimum we determined to need the following: * A project manager to maintain the vision * A data architect to create the data warehouse model * A business integrator to act as the liaison between the project team and the customer * A development team to lead, organize, and develop the ETL process * A validation team to document, test, and provide quality assurance.
When	Did we establish a clear timeline based on scope and resources that met the customer needs and was feasible for the project team to be successful?

Process Planning

Defining Scope	The first step in the testing process is planning. One of the most important and challenging pieces of planning is defining a scope that is scaleable and obtainable. To do this, it is necessary to have a complete understanding of the business needs and project vision. During the requirement development process, the needs and vision must be defined and aligned.
Requirement Definition	Data used in the warehouse originates from multiple source systems. Therefore, defining good requirements can be extremely challenging. Successful requirements are those structured closely to business rules and address functionality and performance. These business rules and requirements provide a solid foundation to the data architects. Using the defined requirements and business rules, the architects create a high-level design of the data model. Once requirements and business rules are available, rough scripts can be drafted to validate the data model constraints against the defined business rules.
Data Dictionary	When there are multiple source systems, there must be some congruent terms and definitions. As part of the requirements process, it is important to create and maintain a data dictionary to capture different terms with similar definitions. This will alleviate ambiguous terms in the requirements.
ETL	The ETL process used needs to fully address whether performance requirements are appropriate. There are several tools available to use for the ETL process. For example, Mercator and Infomatics are two of the more widely used tools for loading warehouses. PL/SQL can be used, depending on the complexity of the data being transformed and loaded. The ETL process could potentially run several times a day. Daily, weekly, monthly, quarterly, and annual production schedules should all be considered when determining when the load of the warehouse should occur. Maintenance of the ETL process is crucial. Regardless whether every performance requirement is met, if the code used to manipulate the data in the ETL process is not easily maintainable, the process is not functional.

Continued on next page

Process Planning cont'd

Validation Staffing

Defining the right resources for the right tasks is a key element to success. It is beneficial to have at least one resource on the validation team with in-depth experience in databases and data models. This resource from the validation team should be involved early and often in the project lifecycle, not just at the end of the project during the testing phase. This will keep the validation team in the loop for requirement, design, and process changes. The staffing should directly reflect the size of the warehouse. Any resources less than two are too few.

Scripting

Creating a scripting process that is right for a data warehouse starts with understanding the source systems, the ETL process, and the warehouse destination. Until the validation team has a solid understanding of how all of this works together, scripting cannot begin. There are at least two approaches that can be considered.

Approach I: follow the data from the source to the target warehouse. This approach validates the data in the source tables (source_tables) is also in the target tables (dw_tables).

Approach II: follow the data from the source through the ETL process and into the target warehouse. Validate the data at each transformation: first the source tables (source_tables), then the staging tables (stage_tables), then the loading tables (load_tables), and finally the destination tables – the warehouse (dw_tables).

Available resources and established timelines tend to drive the approach that is used for the validation process. If time and resources are available, Approach II is the most comprehensive. Since this approach has logical validation points, the validation team can easily determine when and what data has been lost or incorrectly manipulated. Again, the trade-off with this approach is the time and resources necessary to script and execute this test strategy.

Approach I will take less time to script and execute. However, since this approach does not offer logical validation points, once issues are uncovered, it will be more difficult and time consuming to determine where exactly the error occurred in the ETL process.

Validation

Testing Levels

There are several levels of testing that can be performed during data warehouse testing. Some examples,

Constraint testing

Source to target counts

Source to target data validation

Error processing.

The level of testing to be performed should be defined as part of the testing strategy.

Constraints

During constraint testing, the objective is to validate unique constraints, primary keys, foreign keys, indexes, and relationships. The test script should include these validation points.

Some ETL processes can be developed to validate constraints during the loading of the warehouse. If the decision is made to add constraint validation to the ETL process, the ETL code must validate all business rules and relational data requirements.

Depending solely on the automation of constraint testing is risky. When the setup is not done correctly or maintained throughout the ever-changing requirements process, the validation could become incorrect and will nullify the tests.

Counts

The objective of the count test scripts is to determine if the record counts in the source match the record counts in the target. Some ETL processes are capable of capturing record count information such as records read, records written, records in error, etc. If the ETL process being used can capture that level of detail and create a list of the counts, allow it to do so. This will save time during the validation process.

Continued on next page

Validation cont'd

Source to Target

No ETL process is smart enough to perform source to target field-to-field validation. This piece of the testing cycle is the most labor intensive and requires the most thorough analysis of the data. There are a variety of tests that can be performed during source to target validation. Below is a list of tests that are best practices:

Threshold testing – expose any truncation that may be occurring during the transformation or loading of data

For example:
Source: table1.field1 (VARCHAR40):
Stage: table2.field5 (VARCHAR25):
Target: table3.field2 (VARCHAR40):

In this example the source field has a threshold of 40, the stage field has a threshold of 25 and the target mapping has a threshold of 40. The last 15 characters will be truncated during the ETL process of the stage table. Any data that was stored in position 26-30 will be lost during the move from source to staging.

Field to Field

Field-to-field testing – is a constant value being populated during the ETL process? It should not be *unless* it is documented in the requirements and subsequently documented in the test scripts. Do the values in the source fields match the values in the respective target fields? Below are two additional field-to-field tests that should occur.

Initialization – During the ETL process if the code does not re-initialize the cursor (or working storage) after each record, there is a chance that fields with null values may contain data from a previous record.

For example:
Record 125: Source field1 = ***Red*** Target field1 = ***Red***
Record 126: Source field1 = ***null*** Target field 1 = ***Red***

Validating relationships across data sets – Validate parent/child relationship(s)

For example:
Source parent: Purple.
Source child: Red and Blue.
Target parent: Purple
Target child: Red and Yellow.

Continued on next page

Validation cont'd

Error Processing

Understanding a script might fail during *data validation*, may confirm the ETL process is working through *process validation*. During process validation the testing team will work to identify additional data cleansing needs, as well as identify consistent error patterns that could possibly be diverted by modifying the ETL code. Taking the time to modify the ETL process will need to be determined by the project manager, development lead, and the business integrator. It is the responsibility of the validation team to identify any and all records that seem suspect. Once a record has been both data and process validated and the script has passed, the ETL process is functioning correctly.

Conversely, if suspect records have been identified and documented during data validation that are not supported through process validation, the ETL process is not functioning correctly. The development team will need to become involved in finding the appropriate solution. For example, during the execution of the source to target count scripts suspect counts are identified (there are less records in the target table than in the source table). The records that are 'missing' should be captured during the error process and can be found in the error log. If those records do not appear in the error log, the ETL process is not functioning correctly and the development team needs to become involved.

Defect Tracking

A defect-tracking tool should be used. Once a defect is identified, a ticket should be created and assigned to the gatekeeper for further analysis. Once the gatekeeper determines who should be 'working' the defect, the defect is assigned. The resolution could be a clarification and/or may require actual modifications to the ETL process for resolution.

If a breakdown in communication occurs during the tracking of a defect, the validation team could inadvertently fail scripts based on incorrect expected results. For example, a defect is logged regarding bad data. A decision is made to modify the ETL process to scrub the data. The validation team needs to be notified of this change, and subsequently modify the expected results within the appropriate test scripts.

Summary

Lessons Learned

During the validation of our data warehouse, we uncovered some areas for improvement. The following are the most important lessons we learned.

Requirements are one of the keys to success. They are the foundation for understanding and communicating the business needs and desires. If requirements are not comprehensive and complete, implementation will be difficult.

Automating processes whenever possible, will save tremendous amounts of time.

The whole team should share the same tools from the project toolbox. At the end of our project, it was obvious that not everyone had access to the requirements repository, Rational RequisitePro, or to the defect tracking system, Rational ClearQuest. Not having this access slowed down our process and created a level of chaos.

We recommend that everyone work from the same set of documents and requirements.

Project members need to make sure they have the access necessary to view the project "blueprints."

Finally, do not over-simplify the testing efforts that are needed in order for the project to be a success.

Conclusion

Throughout the data warehousing life cycle, it is vital to keep the appropriate focus. Although the immediate goal is to design and build a data warehouse, the big picture goal is to use the data warehouse in one or more applications. In order to maintain the primary goal, the user group, the application team, and the data warehouse team must work together to make sure the end result will receive client acceptance. A lot of "heads-down" effort is required when validating a data warehouse. Plan for it and you will be successful too.