# The Proposal of Data Warehouse Testing Activities

**Pavol Tanuška, Oliver Moravčík, Pavel Važan, Fratišek Miksa**
Faculty of Material Science and Technology
Slovak University of Technology
Paulinska 16, 917 24 Trnava, Slovakia
{pavol.tanuska, oliver.moravcik, pavel.vazan, frantisek.miksa}@stuba.sk

***Abstract****. The lack of the complex Data Warehouse testing methodology (the analysis has been performed and published in former times) seems to be crucial particularly in the phase of the Data Warehouse implementation. The aim of this article is to suggest basic datawarehouse testing activities as a final part of datawarehouse testing methodology.*

*The testing activities that must be implemented in the process of the datawarehouse testing can be split into four logical units regarding the multidimensional database testing, data pump testing, metadata and OLAP testing. Between main testing activities can be included: revision of the multidimensional database scheme, optimizing of fact tables number, problem of data explosion, testing for correctness of aggregation and summation of data etc.*

**Keywords** datawarehouse, test case, testing activities, methodology

## 1 Introduction

### 1.1 Terminology

William Inmon [11] defined Data Warehouse (DW) as a subject-oriented, integrated, stable and time-different data collection supporting the decision-making processes. Data from DW come from non-integrated systems. The system of Data Warehouse requires co-operation of technical resources and program equipment in a heterogeneous environment of information systems to provide analytical information to its users. Basic components of a Data Warehouse environment are as follows: ETL process or data pump, multidimensional database, data marts and metadata. [5]

Verification is the process of ensuring that something – software in this case – meets given specifications. Validation is verification of a product's correctness regarding the user's actual demands. The aim of validation is to record a documented evidence providing a high degree of assurance that, after being set into use, all the parts of equipment will keep working properly. In case of new equipment, validation regards the future requirements, including both, consumer and supplier on one hand, and the process comprising all the activities as they are set into practice on the other hand.

General overview of the validation activities proceeds from the specification through control, design, tests preparation, execution, up to the evaluation of their results. Testing can be therefore considered as one of the basic tools of validation. [9]

The IEEE standard defines a failure as the external, incorrect behavior of a program. Traditionally, the anomalous behavior of a program is observed when incorrect output is produced or a runtime failure occurs. Furthermore, the IEEE standard defines a fault as a collection of program source code statements that causes a failure. Finally, an error is a mistake made by a programmer during the implementation of software. The purpose of software testing is to reveal software faults in order to ensure that they do not manifest themselves as runtime failures during program usage.

### 1.2 Testing process

In the process of software validation, it is necessary to perform the activities providing the high degree of assurance that, after being set into use, all the system parts will keep working correctly. Each software application represents a complex system with its own life cycle, starting with the phases of planning and designing up to the implementation and testing.

Testing is one of the basic components of software. Research and development departments cannot launch software without detecting all the potential defects. The testing aimed at checking

whether software that is being designed performs all the required functions correctly represents one of the most important activities of a software engineer.

Testing (Fig. 1) is a process carried out within the course of program with the aim to detect errors. A well-composed testing task involves a high probability of detecting hidden errors; the test that reveals such an error can be therefore considered successful.

The role of the testing staff is to develop testing tasks systematically, so that the errors of various types are detected at minimum cost and time. Besides detecting the errors, testing demonstrates the functionality of software regarding the given specifications. However, in case of program errors, testing can prove nothing but their existence, without detecting the deficiency caused by them.

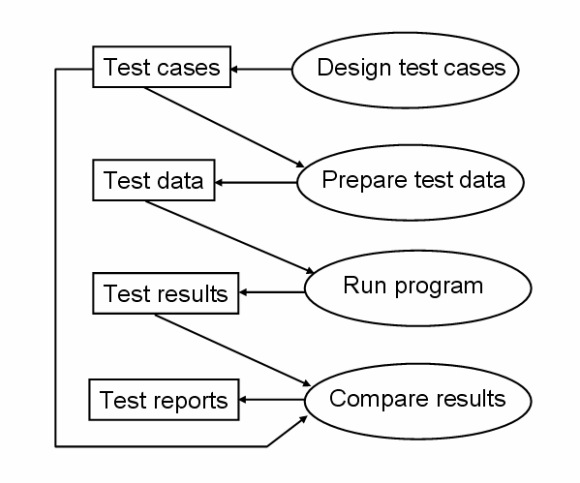


Figure 1. The common testing process

Most of the standards (mostly ISO) and guidelines consider testing inevitable and important, and suggest the procedures and activities accompanying the process. A group of the standards regarding the information systems mostly deals with processes. The processes, including testing, are precisely characterized, divided into activities and further to tasks. The standards do not define the way of how to implement these tasks; however, they may remind us of some of the activities necessary for the correct implementation of the process. The standards are not to replace the systematic management and engineering of the software systems; yet they provide the framework for the unambiguous identification, planning and implementation of processes, activities and tasks. The user can select and build in the way most suitable and cost-effective for the organization.

DIN IEC 56/575/CD standard defines the highest amount of details regarding the process of testing. It states that the aim is to eliminate errors and verify if the system meets the requirements. The test should acquire as much information as possible at minimum costs, and therefore the methods examining several system indicators at the same time are preferred.

There are many testing methods and approaches available [7], yet individual standards do not go into details, providing thus a space for various desinformation.

As for the Data Warehouse design, it should be stated that the existing standards and guidelines do not cover the activities relevant for building a Data Warehouse.

There is no particular procedure or activity related with the process of the multidimensional database design, ETL process or optimization of scripts for OLAP reports.

The following part therefore deals with the proposal of basic datawarehouse testing activities as a final part of datawarehouse testing methodology. Other parts of methodology were published in [1], [2], [3].

# 2 The testing activities proposal

The testing activities that must be implemented in the process of the Data Warehouse testing can be split into a few logical units. In this article I will try to propose testing activities for the multidimensional database, data pump and DW system testing.

## 2.1 Multidimensional database testing

When splitting it further, we get the following activities:

**Revision of the multidimensional database scheme in design phase**

To achieve the best efficiency of SQL statements possible, it is necessary to keep to the following rules:
- each foreign key of the fact table should be indexed,
- 1:n relation between the fact table and dimension tables must be always kept to,
- attribute relation 1:1 between hierarchy levels and their dependent dimension attributes must be always kept to,
- make sure that the columns of each hierarchy level (fact table) are NOT NULL and that hierarchical integrity is kept to,
- columns of the hierarchy level cannot be associated with more than one dimension,
- structure of the columns in the dimension table should be in denormalised shape,
- hierarchy levels cannot be mutually interconnected, recourse must not occur.

**Testing the multidimensional model by a user**

Prior to the implementation of Data Warehouse proposal, it is essential that the user tests the model so that to confirm, that the model meets the user's requirements and that the user is able to comprehend the model. However perfect it may be in meeting all the requirements, the model is ineffective if the user does not understand it and cannot therefore access the data correctly and effectively.

**Optimization of number of fact tables**

In designing a multidimensional database, it is very important to decide if to implement a Data Warehouse as a whole or to start with the implementation of a smaller Data Warehouse and data marts. Regarding the efficiency and the speed of response of the whole system, it is more favorable to propose a smaller number of fact tables corresponding to a data mart or a smaller Data Warehouse. The current trend in building the Data Warehouses is to design a smaller compact Data Warehouse and several satellite data marts.

**Problem of data explosion**

The basic problem is that the size of a database is not equal to the amount of information stored in it. A database explosion is primarily due to high data sparsity and the high number of derived members and aggregated dimensions in the consolidation hierarchies. This happens with the design which does not consider a higher number of fact and dimensional tables. The problem can be eliminated by designing several smaller compact data marts.

## 2.2 Data pump testing

When splitting it further, we get the following activities:

**Testing ETL processes**

The testing of these processes (sometimes called data pump) represents a very important step in the process of building a Data Warehouse. It is the most complex and demanding part in building a Data Warehouse. Testing itself comprises the testing of each script, program and modules, modules' integrity, as well as consistency of the data being transformed into the Data Warehouse.

The testing stage of ETL processes, shown on Fig. 2 involves also the following activities:

- Testing for correctness of aggregation and summation of data

In this testing stage, it is necessary to check the correctness of forming the data aggregation. After the reverse transformation, all the aggregated data being filled into the Data Warehouse should regain their original values.

- Check for reversibility of data from Data Warehouse into OLTP systems

This kind of test is closely connected with the previous activity. It is a control process of reverse transformation of the aggregated and summarized data into the operational databases.

- Check of distributed processing

With distributed operational systems, it is necessary to test the data for the condition of recency, i.e. if the replic in question will be transformed into the warehouse in the correct time horizon.

- Testing data types and metric units

It is the process of testing that must be implemented in the stage of the ETL process design.

- Testing of relationship

In the transformation of heterogeneous files into a Data Warehouse, it is necessary to check the correctness of the design of relationships which were not carried out (e.g. in DBF and XLS files transformations).

- Testing the process of updating

It is necessary to propose the correct time of updating the data in a Data Warehouse from operational systems. The updating depends on more factors, such as the efficiency of a Data Warehouse, rate of the new data processing or volume of the new data.

- Storing converted data

The storage of the data from related systems, such as various files, which should be converted into a defined format, must be tested separately supposing that the individual steps of converting and storing run correctly.

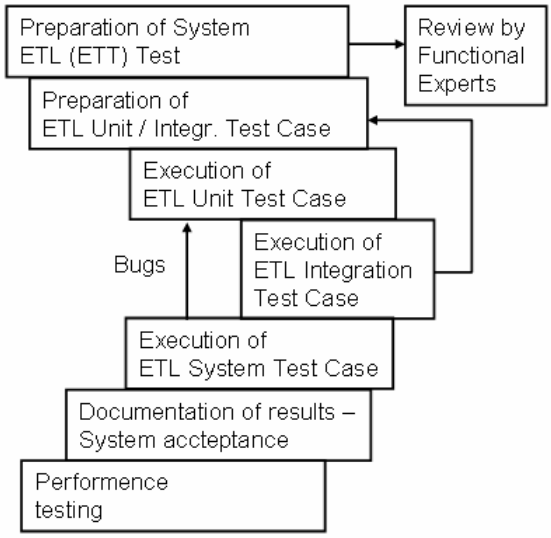


Figure 2. ETL testing process [8]

- Revision of converted data

Once stored in a system, the data must be checked for multidimensionality, such as consistency of data types, number of excluded lines, the reason for exclusion, as well as any logical errors in the process, which might result into the logical non consistency of data.

Besides, ETL process requires the strategic access, i.e. the same care of life cycle common in case of any software product's development. It is therefore necessary to employ all the standard testing activities, including testing metrics.

**User-Triggered vs. System triggered**

Most of the production system testing is the processing of individual transactions, which are driven by some input from the users (Application Form, Servicing Request.). There are very few test cycles, which cover the system-triggered scenarios (Like billing, Valuation.) In Data Warehouse, most of the testing is system triggered as per the scripts for ETL ('Extraction, Transformation and Loading'), the view refresh scripts etc. Therefore typically Data-Warehouse testing is divided into two parts: 'Back-

end' testing where the source systems data is compared to the end-result data in Loaded area, and 'Front-end' testing where the user checks the data by comparing their MIS with the data displayed by the end-user tools like OLAP [4].

**Selecting a relevant metric**

There is nothing like a universal measure to be used or employed for measuring perfectly the quality concept of a suggested system. The decision regarding the choice of a metric is quite complicated, as various users prefer various software attributes.

One of the major problems in the selecting the relevant metric is that the metric is associated with a small number of measurable criteria.

**Load tests**

Each modul of a Data Warehouse must be tested for the volume of data to be used, and with real value of data. The volume of data stored in a warehouse is an important factor influencing the racionalisation of efficiency. If the limits are set, the modul must be tested for the behavior of the limits inside and outside the boundaries.

**Volume of Test Data**

The test data in a transaction system is a very small sample of the overall production data. Typically to keep the matters simple, we include as many test cases as are needed to comprehensively include all possible test scenarios, in a limited set of test data. Data Warehouse has typically large test data as one does try to fill-up maximum possible combination and permutations of dimensions and facts [4].

## 2.3 Data Warehouse system testing

**Testing the data back up and recovery**

Prior to starting a Data Warehouse, the strategy of data back-up and recovery must be configured. These tests must be executed with the volume of data equal to the one in the real system, in order to examine possible effects in the process of the full data recovery.

**Testing the on-line time response**

The testing should take place via iniciating the pre-defined demands simulating the daily-anticipated efficiency of a Data Warehouse, while testing the time response.

**Testing the time shift (up to date)**

It is necessary to pay attention to the possible time shift in processing the data from Data Warehouse to OLAP server.

**Testing the access to data**

In this stage, it is necessary to test ad-hoc prepared reports and fixed reports. This requires that the reports' contents and requirements be defined correctly.

**Sequence testing**

Data is transferred into the Data Warehouse manually, i.e. the loading activities are launched manually, step by step; i.e. the result of loading is also checked step by step.

**Testing the complexity**

It is tested whether the whole system works correctly together with all the tasks planned, automated transfer, starting the work, handling the errors, converting the data etc.

**Testing the batch processing response**

The testing for the efficiency of a batch processing should be implemented by simulating the batch in the system loaded with real data with real infrastructure and operating at the same time as a real system.

**Possible number of testing scenarios**

If a transaction system has hundred different scenarios, the valid and possible combination of those scenarios will not be unlimited. However, in case of Data Warehouse, the combinations can possibly test is virtually unlimited due to the core objective of Data Warehouse is to allow all possible views of Data. In other words, 'there is no possibility fully to test a Data Warehouse. [4].

**Limited Data Warehouse test Data**

This involves feeding limited transactions in the source systems (typically less than few thousands for each data-mart'). This should ideally take care of key scenarios in terms of different Transformation logics. Say, if Transformation is doing some de-duping, place couple of duplicate cases. For customer dimension (say) you can have customers of different ages, income groups etc. After these transactions have been processed by the source systems, the entire processing is conducted and results are checked at each interim stage and also in the end user tools [4].

**Metadata administration**

It is necessary to correctly define the metadata in the beginning of building a Data Warehouse, and administer them in the course of running the Data Warehouse.

Metadata enable us to exactly identify the errors which may occur in the process of using a warehouse in practice, as well as to determine when individual processes such as ETL can be started.

Prior to testing, all the related documents, approved test specifications and testing procedures must be available. Each test must be executed by an authorized person.

**Testing time consistency**

One of the most important factors with Data Warehouses is time consistency allowing the user to acquire real responses to their SQL statements. It is necessary therefore to decide on the right granularity closely related with the time consistency.

# 3 Conclusion

The testing phase as one of the stages of DW development lifecycle is very important, since the cost depleted for the elimination of a potential error or defect in a running Data Warehouse is much higher. A Data Warehouse as well as an information system

can be physically correctly tested only when the working database is loaded.
The aim of this article has been to suggest basic datawarehouse testing activities as a final part of datawarehouse testing methodology.

# 4 Acknowledgments

This contribution as a part of the project No. 1/4078/07 was supported by VEGA, the Slovak Republic Ministry of Education's grant agency.

# References

[1] Tanuška, Pavol - Moravčík, Oliver - Važan, Pavol - Miksa, František: **The Proposal of the Essential Strategies of Data Warehouse Testing**.
In: 19th Central European Conference on Information and Intelligent Systems - CECIIS : Conference Proceedings. Croatia, Varaždin, September 24-26, 2008. - Varaždin : University of Zagreb, 2008. - ISBN 978-953-6071-04-3. - S. 63-67

[2] Tanuška, Pavol - Verschelde, Werner - Kopček, Michal: **Proposal of a Data Warehouse Test Scenario.** In: Ecumict 2008 : Proceedings of the Third European Conference on the Use of Modern Information and Communication Technologies. Gent, Belgium, 13-14 March 2008. - : Nevelland v.z.w., 2008. - ISBN 9-78908082-553-6. - S. 403-409

[3] Tanuška, Pavol - Schreiber, Peter - Zeman, Jaroslav: **The realization of Data Warehouse test scenario.** - 1/4078/07.
In: Infokommunikacionnyje technologii v nauke, proizvodstve i obrazovanii. (Infokom-3) Časť II : III. meždunarodnaja naučno-techničeskaja konferencija. 1-5 maja 2008, Stavropoľ. - Stavropoľ : Severo-Kavkazskij Gosudarstvennyj Techničeskij Universitet, Russia, 2008. - S. 101-107

[4] BiPM Institute - **Building Intelligent and Performing Enterprises**. April 2009., Available on web: http://www.bipminstitute.com/data-warehouse/

[5] Ponniah, P. **Data Warehouse Fundamentals – Comprehensive Guide**. London: John Willey and Sons, 2001

[6] Tanuška, Pavol - Moravčík, Oliver - Važan, Pavol - Miksa, František: **The Proposal of the Essential Strategies of Data Warehouse Testing.** - Vega 1/4078/07.
In: 19th Central European Conference on Information and Intelligent Systems - CECIIS : Conference Proceedings. Croatia, Varaždin, September 24-26, 2008. - Varaždin : University of Zagreb, 2008. - ISBN 978-953-6071-04-3.

[7] Cooper, R., Arbuckle, S.: **How to Thoroughly Test a Data Warehouse**. STAREAST, 2002.

[8] Asimkumar, M.: **Testing a Data Warehouse Application**, Taken from http, Published, 2003.

[9] Gregory M. Kapfhammer: **Software Testing. The Computer Science Handbook**, Publisher: CRC Press. June, 2004, Available on web: http://www.cs.allegheny.edu/~gkapfham/

[10] Strémy, Maximilián - Eliáš, Andrej - Važan, Pavol - Michaľčonok, German: **Virtual Laboratory Implementation.**
In: Ecumict 2008 : Proceedings of the Third European Conference on the Use of Modern Information and Communication Technologies. Gent, Belgium, Nevelland v.z.w., 2008. - ISBN 9-78908082-553-6.

[11] Inmon, W.H. **Building the Data Warehouse**. New York, USA: John Willey and Sons, 2002 ISBN
0-471-08130-2