Data Warehouse Testing- An Iteration Based Methodology

Payal Pahwa¹, Renu Miglani²

¹Bhagwan Parshuram Institute of Technology, JPU University, Delhi, India
pahwapayal@gmail.com

²GVM Institute of Technology & Management, DCRUST Murtial, Sonepat, Haryana, India
rnkakkar@gmail.com

Abstract

Now a days, Data Warehouse (DW) is a critical business application and defects in it leads to improper decision making. A defect free DW implementation is a dream comes true for any organization. Therefore testing is an indispensable phase of any DW development process to fix these tribulations. A defect traced at later stages in DW development life cycle exponentially increases the cost of testing because the additional business cost of using incorrect data to make crucial business decisions also compounds to it. This paper proposes an Iterative DW testing life cycle and multiple stages of Incremental Integration testing to detect the defects as early as possible. It also explores the phases of DW testing and provides a brief description of related work.

Keywords: DW Testing, DW Testing Life cycle, Unit Testing, Integration Testing, Regression Testing.

I. INTRODUCTION

A Data Warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process [1]. It is a composite and collaborated data model that captures the entire data of an organization [2]. Most of the organizations run their businesses on the basis of collection of data for strategic decision-making to take a competitive edge. The quality of real world data that is fed to a DW is of major concern that comes from disparate sources. The presence of any inconsistency, errors, flaws or bug in the data or system may affect strategic decision or even halt the system. Therefore, testing is required at the beginning as well as in every phase of DW development process. DW testing is an activity that is intended for evaluating quality of a DW and also for improving it, by identifying defects and problems. As with any other piece of software a DW implementation undergoes the usual cycle of unit testing, system testing, integration testing, regression testing and acceptance testing. However, unlike others there are no off the shelf testing products available for a DW [4]. Testing should start at the requirement phase and continue throughout the development lifecycle to detect as many defects as possible in the system early, to reduce the instances of defects converting into failures.

II. TESTING DATA WAREHOUSE

Data Warehouse is the main repository of any organization's historical data. It contains the material and information for management's decision support system [3]. DW testing is an activity that is intended for evaluating quality of a DW and also for improving it, by identifying defects and problems. As with any other piece of software a DW implementation undergoes the usual cycle of unit testing, system testing, integration testing, regression testing and acceptance testing. However, unlike others there are no off the shelf testing products available for a DW [4]. Testing should start at the requirement phase and continue throughout the development lifecycle to detect as many defects as possible in the system early, to reduce the instances of defects converting into failures.

III. RELATED WORKS

Data Warehouse testing is a novel and comparatively adolescent field so only a few works discuss the issues raised by testing in the Data Warehousing context.

[3] Describes ETL testing and the main challenges, phases and goals of DW testing and also discriminate between the retrospective and prospective testing. The paper also suggests testing activities related to incremental load using
assembled data. It proposes some basic testing rules that should follow to reduce the high testing cost and shares a real world example faced while performing DW testing.

[5] Defines a framework for data mart testing and introduces a number of data mart specific testing activities to frame them within a reference design methodology. This paper also differentiates between generic software system testing and Data Warehouse testing. It also proposes a timeline for testing and classifies the testing activities. It discusses the eight type of tests that best fit the characteristics of DW.

[6] Proposes an approach that applies the business rules to the data to develop testing procedures that check the accuracy of entire data set. It includes data verification which tests the quality of data populated into target tables and this methodology will eventually lead to a more effective implementation. It suggests testing methodology as V-Model to face the challenges related with DW testing.

[8] Mentions eight classical mistakes in Data Warehouse testing using only artificial data without closely involving end users moreover unit testing, system testing, acceptance testing, and performance testing are proposed as the main testing steps. It focuses on testing the reports rather than data.

[9] Talks about the main characteristics in Data Warehouse testing, different roles required in the testing team, and also the types of testing each role should hold to improve the quality of DW. It also suggests some tests and testing tools to facilitate testing and problem correction.

[10] Explains the process of Data Warehouse testing consist of requirement testing, unit testing, integration testing and user acceptance testing and also discuss some major concerns regarding DW testing. It suggests testing for data contains in the dimensions and fact tables as well as the data presented to the presentation layer. It proposes some standard tests for DW that should be carried as part of testing for every project.

[11] Recommends a fundamental set of attributes like the test purpose, its performance requirements, its acceptance criteria, and the activities for its completion for a Data Warehouse test scenario based on the IEEE 829 standard.

[12] Describes general Data Warehouse structure and background, as well as specific situations encountered during the testing of data mart for a large research department of a pharmaceutical company. It also explains DW testing along with constraint testing, source-to-target testing, and validation of error processing procedures.

[13] Suggests a detailed list of testing categories and also distinguishes between testing OLTP and OLAP systems. It also proposes various types of test data to be used for testing and itemises probable test scenarios.

IV. PHASES IN DATA WAREHOUSE TESTING

From a methodological point of view, the testing phases are [3] & [5]

- **Form test plan:** Test plan describes the testing level, sequence, strategy, environment and the items to be tested.
- **Design test cases:** The aim of test case design is to examine every component, module and the entire system using extensive series of conditions.
- **Setup test environment:** Test environment may be natural or contrived. Contrived environment is set up by the tester in a place like laboratory whereas the natural environment is the system's place of work.
- **Prepare test data:** Test data may be mock or actual. Mock data can be design to check all transactions, logic and the limit of system by providing all combinations of values and extreme values during testing process whereas actual data may be inclined towards typical values. Test data generation tools or customized tools can also be used to generate test data.
- **Execute tests:** Specific process is followed to test each item at each level with appropriate test suite. Test execution log will be maintained to track each test along with its result.
- **Compare test result with requirement document:** The result of each test execution will be measured against requirement criteria specified in Requirement Specification document to evaluate the system.
- **Test Report Generation:** Test reports should be created at distinct points during the testing process which contains
summarised results of testing and used as a verification document for analysis.

As with any other piece of software a DW implementation undergoes the natural cycle of Requirement testing, Unit testing, Integration testing, System testing, Regression testing and Acceptance testing [6]. Defects perceived at any stage need system modification to fix them and this may require some of the previous stages of testing process to be repeated.

- **Requirement Testing**: Requirement specification document specifies the functionality to be achieved by the system so it must be reconsidered along with the user of the system. It is used as traceability matrix to ensure the system functionality. Requirements must be reassessed for completeness, feasibility, traceability, consistency, usability, certainty and nonexistence of contradictions. Interface review is done to check the usability, practicability and traceability with the design as well as according to the specification document.

- **Unit testing**: This is a white-box testing executes at developer site in which each and every component of a developed module is tested individually. Any change in the functionality of a module will follow the retesting of unit and the process is repeatable until verification. Tests are executed to verify the individual procedures, jobs, mappings and reports.

- **Integration testing**: The process of combining and testing the components for the tribulations arise from the interaction among them. The components will be integrated in either top down or bottom up manner.
  - **Top down Integration**: The formation of the system outline will be done first and then it will
be populated with components.

- **Bottom up Integration** - The infrastructural components will be incorporated to attach functional components.

Integration testing considers the compatibility of the DW modules in upward and downward streams. It is done in incremental order consisting of multiple stages after incorporation of every component to ensure that they work together. The process of Incremental Integration testing continues until the full functional system is obtained as required. The testing process concentrates on detection of interface errors by thoroughly exercising these interfaces [7].

- **Regression testing**: Data Warehouse is the best model of incremental design and it is not implemented only one time. Whenever organisation requirements are enhanced according to business needs and user feedbacks, an enhancement is made to it. Whenever a new module is attached to it, all the functional tests are executed again to check the non corruption of existing functionality by comparing the new results with the older ones. Regression testing verifies that the system continues to function correctly after being modified.

- **System testing**: This testing executes at developer site in which the whole DW is tested together regarding functionality. The aim of system testing is to confirm that the required specifications are implemented correctly and all components work together. System is tested by using test data to find the errors and exceptions. In system testing test cases are based upon design and specification document.

- **Performance testing**: The system is monitored and its performance is recorded at low, regular and high level. The performance is judged while transforming and loading the large amount of data. It incorporates resource usage, throughput, response time and waiting time to evaluate the performance. The system must be proficient to return results quickly without slowing down other resources.

- **Acceptance testing**: It is a black box testing done prior to transition and project completion. The system is tested with real data and the actual users of the DW validate the system by ensuring the result according to expectations. Acceptance testing may detect problems related with specifications and finds that system functions do not fulfil the user requirements, in such a case the whole DW development process will be repeated from scratch. A system cannot be considered to satisfy its specifications until it has passed an acceptance test.

### VI. Conclusion

Data Warehouse testing is a crucial phase of any DW development process which is done with the intention of finding the error in DW as early as possible. An error in a DW traced at a later stage may cause unpredictable results and increased the cost drastically. Defects perceived at any stage need system modification to fix them and this may require DW testing cycle to be repeated from a specific previous level. This paper explains the Iterative DW testing life cycle and stages of Incremental Integration testing along with pictorial presentation. It also discusses DW testing, explored phases of DW testing as well as provides...
a review on related work.

REFERENCES


