A CRITICAL STUDY OF REQUIREMENT GATHERING AND TESTING TECHNIQUES FOR DATAWAREHOUSING

KULDEEP DESHPANDE¹, Dr. BHIMAPPA DESAI²

¹(Ellicium Solutions, Pune, Maharashtra)
²(Capgemini Consulting, Pune, Maharashtra)

ABSTRACT

In light of high cost and higher rate of failure of Datawarehousing projects, it becomes imperative to study software processes being followed for Datawarehousing. In this paper we present a survey of literature for Datawarehousing requirement gathering and testing. This paper has analyzed drawbacks of traditional techniques for requirement gathering and testing of Datawarehouse. We have reported areas where more research needs to be focused. Using text analytics technique called “word cloud”, we have analyzed main areas being researched and shown areas that need more focus. This paper can give a direction to future research in the areas of Datawarehouse requirement gathering and testing.

Keywords: Datawarehouse, Lifecycle, Requirements, Techniques, Testing.

I. INTRODUCTION

A data warehouse is a subject oriented, integrated, time-variant, and nonvolatile collection of data that supports managerial decision making [8]. Datawarehousing projects are costly and risky as reported by many researchers. Long development cycles and large expensive costs are typical of a Datawarehouse development project. Rate of failure of datawarehousing projects is very high. Hence it becomes imperative to study various software processes being followed for Datawarehousing, advantages and limitations of various process and alternatives for existing software processes.

This work is organized as follows. In section 2 we briefly discuss software development lifecycle stages for datawarehouse development proposed by various authors. In
section 3 we discuss various requirement gathering techniques. We also discuss about coverage of nonfunctional requirements in this section. Limitations of traditional requirement gathering techniques are discussed in section 4. Next in section 5 we discuss various testing techniques proposed in literature. In section 6, we discuss limitations of traditional testing techniques. Section 8 summarizes our conclusions which include recommendations for future research.

II. DATAWAREHOUSE DEVELOPMENT LIFECYCLE STAGES

In this section we define a set of activities that constitute datawarehouse development process.

As per a study of Datawarehousing process maturity [2], a data warehousing process can be viewed as a data production process that includes sub processes such as business requirements analysis, data design, architecture design, data mapping, ETL design, end-user application design, data quality management, business continuity management, implementation, and deployment. We have modified this categorization and have categorized datawarehouse development tasks proposed by various studies in 5 categories: Requirements, Design, Development, Implementation and Project management.

Based on various studies [16], [12] and [1] following is a list of activities that constitute various phases of Datawarehouse development lifecycle:

| TABLE 1: Classification of Datawarehouse development lifecycle activities |
|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
| Requirements | • Requirement Analysis | • Preliminary study | • Business Requirement definition |
| Design | • Analysis and Reconciliation, | • Technical requirements , | • Technical architecture design, |
| | • Conceptual design, | • Technical design, | • Product selection and installation, |
| | • Workload refinement, | | • Dimensional modeling, |
| | • Logical design, | | • Physical design, |
| | • Data staging design, | | |
| | • Physical design, | | |
| Development | • Implementation | • Weaving, | • Data staging design and development, |
| | | • Code generation, | • Application specification and development |
| | | • Code completion and tests | |
| Implantation | No task specified | • Deployment | • Deployment, |
| | | | • Maintenance, growth |
| Project Management | No task specified | No task specified | • Project Planning |
Information requirements gathering for data warehouse systems differ significantly from requirements analysis for conventional information systems. Same is applicable to DW testing. In this paper we focus on these two critical phases of DW development lifecycle.

III. REQUIREMENT GATHERING FRAMEWORKS FOR DW PROPOSED IN LITERATURE

Requirement analysis for Datawarehouse is one of the widely researched areas in Datawarehousing. Ralph Kimball and Bill Inmon, two pioneers in Datawarehousing have extensively covered the topic of requirement gathering and analysis in their work.

Ralph Kimball has elaborated extensively methods for requirement gathering for Datawarehouse in his book – “The Datawarehouse Lifecycle Toolkit” [1]. Requirements determine what data must be available in the data warehouse, how it is organized, and how often it is updated. Business users and their requirements impact almost every decision made throughout the implementation of a data warehouse [1]. Kimball has proposed an overall approach for requirement gathering for Datawarehouse with emphasis on interviewing of business users, data auditing and consensus building [1]. Thus Ralph Kimball’s approach towards requirement gathering in Datawarehouse is a user focused approach.

Bill Inmon has emphasized on uncertain nature of requirements in Datawarehouse. He has mentioned that requirements are the last thing to be discovered in a Datawarehouse. Requirements cannot be discovered until the Datawarehouse has been built and is in use for some time [8]. According to Inmon “Data warehouses cannot be designed the same way as the classical requirements-driven system. On the other hand, anticipating requirements is still important. Reality lies somewhere in between” [8]. Inmon has proposed use of ‘Zachman Framework’ for requirement gathering of Datawarehouse [8]. Thus, Bill Inmon’s approach toward requirement gathering in Datawarehouse is a source system driven approach. These 2 approaches towards requirement gathering have evolved into User Driven and Data Driven approaches.

3.1 User driven requirement gathering approach

In User Driven approach for requirement gathering, focus is on understanding data needs of end users. The definition of the business requirements determines the data needed to address business users’ analytical requirements [1]. User driven approach makes use of techniques such as interviewing end users to understand their data needs, analysis of existing reports and workshops with business stakeholders.

User driven approach results in high level of end user involvement in DW design process. One of the main issues faced in many Datawarehouses is less or no usage of DW by end users. By involving end users at an early stage in the requirement gathering, acceptance of the DW is increased [3].

However, DW designed using user driven approach may face risk of obsolesce if the requirements are based on discussions with only an incorrect sample of end users OR if opinions by end users do not express organizational data needs. Also, rather than forming an Enterprise wide Datawarehouse, this method may result in forming an aggregation of existing data marts and may result in existing isolated data marts being replicated in the DW.

3.2 Goal driven approach of requirement gathering

Goal driven approach for DW requirement gathering focuses on capturing business goals that the DW is expected to serve. These goals are then converted into a good ETL / DW
design. In [5] the authors have proposed a goal-oriented approach to requirement analysis for data warehouses. The paper has proposed requirement modeling in 2 phases i.e. Organizational modeling and Decision modeling. Conceptual design is proposed as a next phase.

3.3 Supply driven approach of requirement gathering

Supply driven approach to requirement gathering focuses on analysis of operational data sources to analyze available data for loading into the datawarehouse. In this approach involvement of end users is limited to deciding which data is relevant to decision making. However, in general the approach is to load all possible data into the Datawarehouse and let the end users use data that they are interested in. Supply driven requirement gathering is mainly driven by source system experts who provide their inputs about structure and content of source systems. This approach ensures availability of data and risk of obsolesce of data because of incompleteness of datawarehouse is less [3]. Datawarehouse designed by this approach quickly grows to a complex size with increase in number of source systems. Also lack of end user involvement in requirement gathering and design may result in less acceptance of the datawarehouse by the end users.

3.4 Combinational approaches

During last few years there has been a growing realization that user driven and supply driven approaches in isolation cannot lead to a successful datawarehouse. In [3] authors conclude that the adoption of a “pure” approach is not sufficient to protect from its own weaknesses. Authors believe that coupling a data-driven step with a demand-driven one can lead to a design capturing all the specifications and ensuring a higher level of longevity as well as acceptance of the users. The authors stress on formalism in requirement gathering when business users are involved.

In [7] the authors present a data modeling methodology in data warehousing which integrates three existing approaches normally used in isolation: goal-driven, data-driven and user driven. The authors have demonstrated various stages of proposed methodology using a case study. The goal-driven stage produces subjects and KPIs of main business fields. The data-driven stage obtains subject oriented data schema. The user-driven stage yields business questions and analytical requirements. The combination stage combines the triple-driven results. In [4] authors have proposed key principles for requirement analysis phase of a Datawarehouse project. Based on these principles and findings from a collaboration project, the authors have proposed a demand driven methodology. The synchronization of information demand and information supply is done in a two-step, end users are involved in the specification process at several occasions.

Devising a design methodology is almost useless if no CASE tool to support it is provided [11]. In [11] authors have proposed a CASE tool, WAND based on bottom up approach for building Datawarehouse. Based on this approach a CASE tool has been designed. Thus the focus of research in the area of requirement gathering for datawarehouse has been shifting from pure ‘supply driven’ and ‘user driven’ approaches in early years to mixed approaches and more practical view towards requirement gathering.

3.5 Model Driven Approach to DW requirements and design

Model Driven Architecture is an approach for system specification and interoperability based on use of formal models. In [10], authors have described how MDA and CWM (Common Warehouse Metamodel) can be used for requirements gathering and
design of Datawarehouse. Similarly in [12], a datawarehouse framework and Unified process has been proposed for development of datawarehouse using model driven architecture.

3.6 Nonfunctional requirements for Datawarehouse

With respect to DW systems, functional requirements specify what information the DW is expected to provide and non functional requirements specify how the information should be provided [3]. In [15] authors have extended NFR framework proposed by Chung, Nixon et al to define catalog of major DW NFR types. NFRs have been defined as soft goals and correlation among them is identified. NFRs identified by authors include: performance, security, multidimensionality, user friendliness. Proposed framework has been applied to a DW for Brazilian government. In [14], authors have defined ‘Datawarehouse Extended NFR framework’ (DW-ENF) by extending NFR framework. In this framework, they have proposed set of NFR types and operationalization catalogs for DW.

We studied the approach followed by major studies of DW requirement gathering for their treatment of NFRs. Below table lists treatment of NFR in these approaches:

<table>
<thead>
<tr>
<th>Reference</th>
<th>Type of approach for DW requirements</th>
<th>Focus on NFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4, 9</td>
<td>Demand driven requirement analysis for DW</td>
<td>No coverage for NFR</td>
</tr>
<tr>
<td>5</td>
<td>Goal oriented requirement analysis for DW</td>
<td>No coverage for NFR</td>
</tr>
<tr>
<td>10</td>
<td>Model driven approach for DW design</td>
<td>No coverage for NFR</td>
</tr>
<tr>
<td>12</td>
<td>Integrated / mixed approach for design</td>
<td>No coverage for NFR</td>
</tr>
<tr>
<td>3</td>
<td>Literature survey of DW requirement analysis studies</td>
<td>Concludes that DW requirement approaches have neglected NFRs</td>
</tr>
</tbody>
</table>

Thus, in the traditional DW requirement gathering approaches listed above, coverage for NFRs has been completely neglected.

IV. LIMITATIONS OF TRADITIONAL REQUIREMENT GATHERING TECHNIQUES

Despite of extensive research on requirement gathering techniques for datawarehouse in the literature, there are certain gaps that need to be addressed.

1. Integration of requirement gathering with physical design - Integration of requirements with subsequent development phases is an important characteristic of any requirement gathering technique [4]. Kimball has stressed that requirements should determine not just what data should go into Datawarehouse, but also how it is organized and updated [1]. However, none of the existing literature mentions how DW requirements can be linked to ETL design, metadata design. Most of the focus in existing literature is on linking requirements with data modeling. In general we found very little focus on requirement gathering for physical design of Datawarehouse from end users’ perspective. There is a need to view requirement gathering techniques as a part of entire development lifecycle than as a standalone activity.
2. **Lack of focus on Model driven approaches for requirement gathering** - In recent years, there has been extensive research in the area of Model Driven Architecture for Datawarehouse. However, linking MDA with traditional user driven and supply driven approaches of requirement gathering has not been given due attention.

3. **Lack of empirical evidence on suitability of requirement gathering techniques** - Most of the literature focuses on merits and demerits of requirement analysis methods. Very few of them describe experience of implementing various approaches in real life projects. Especially relationship of approach for requirement gathering with success of DW is not given due attention. [7] Describes a case study of implementation for an Insurance organization. Such a study can be an important guide for DW practitioners.

4. **Need for development of CASE tool for DW requirement gathering** - Devising a design methodology is almost useless if no CASE tool to support it is provided [11]. Literature lacks extensive examples of CASE tools.

5. **Very little and isolated focus on nonfunctional requirement** - In the traditional DW requirement gathering approaches, coverage for NFRs has been very rare. Literature lacks empirical studies on treatment of NFRs in real life projects as well as in lab environment. Majority of the functional requirement gathering techniques and non functional requirement techniques for Datawarehouse have been treated separately in literature. Relationship between functional and non functional requirements of a Datawarehouse needs to be explored. CASE tools for requirement gathering of Datawarehouse are focused on functional requirement gathering only [11] and there is a scope to enhance these CASE tools with steps for NFRs.

### V. DATAWAREHOUSE TESTING TECHNIQUES

Data warehouse testing is relatively new field and lacks process maturity and knowledge sharing [21]. DW testing is different from generic software testing in many ways. DW testing involves a huge data volume [16]. The key to data warehouse testing is to know the data and what the answers to user queries are supposed to be [16]. Testing both initial and incremental loads of data is a key difference between DW testing and IT systems testing [17]. High number of scenarios in DW testing is an important difference between general purpose testing and DW testing [23], [16]. [25] Has presented a detailed review of reasons why DW testing is different from software testing.

#### 5.1 Importance of early testing

Advancing an accurate test planning to the early projects phases is one of the keys to success [16]. If the bug is detected at the later stage of testing cycles, it could easily lead to very high financial losses to the project [17]. Early participation of testers in requirement and design activities has been advocated in [21]. A good understanding of data modeling and source-to-target data mappings help equip the quality assurance (QA) analyst with information to develop an appropriate testing strategy. During the project’s requirements analysis phase, the QA team must work to understand the technical implementation [26].
5.2 What needs to be tested?

Because of reasons cited, DW testing is different and requires different types of components to be tested compared to traditional software testing. Various authors have proposed approaches to categorize activities in DW testing.

**TABLE 3: Classification of testing methods by various studies**

<table>
<thead>
<tr>
<th>Reference study</th>
<th>Categories of testing activities proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Comprehensive Approach to Data Warehouse Testing [16]</td>
<td>ETL procedures, Database, Front end</td>
</tr>
<tr>
<td>The Proposal of Data Warehouse Testing Activities [23]</td>
<td>Multidimensional database testing, Data pump testing (ETL), Data Warehouse system testing *</td>
</tr>
<tr>
<td>Classification of Datawarehouse testing approaches [25]</td>
<td>Query level, Functional level, Quality Assurance level, Engineering level, Adaptability level</td>
</tr>
</tbody>
</table>

We extend the classification framework proposed by various studies and propose following classification for testing activities:

1. Business requirement testing – Testing whether requirements expressed by business users have been met.

2. ETL testing – Testing accuracy of data movement from source systems to Datawarehouse as per specifications.

3. Datawarehouse database testing – Testing the database performance at normal and stressed workloads. This category also includes tests to verify data quality in the DW.

4. System testing – Testing entire integrated datawarehouse system to evaluate ability to meet functional as well as nonfunctional requirements.


6. Reporting / Front end testing – Testing if reports / OLAP system provide functionally correct data access to end users

Based on this classification framework following is a list of activities that constitute that can be categorized under each category:
TABLE 4: Proposed framework of classification of Datawarehouse testing activities.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Data Warehouse Testing an Exploratory Study [22]</th>
<th>The Proposal of Data Warehouse Testing Activities [23]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Requirement testing</td>
<td>Testing the business requirements</td>
<td></td>
</tr>
<tr>
<td>ETL testing</td>
<td>Verify Slowly Changing Dimensions Verification of performance of ETL batch loading Data transformation</td>
<td>Testing ETL processes Load tests Volume of Test Data</td>
</tr>
<tr>
<td>Datawarehouse database testing</td>
<td>Ability to perform in high volume data Data completeness Test for relevancy Data quality</td>
<td>Revision of the multidimensional database scheme in design phase Testing the model by a user Optimization of number of fact tables Problem of data explosion</td>
</tr>
<tr>
<td>System testing</td>
<td>Test for daily, weekly or monthly batch runs Recoverability</td>
<td>Testing the data backup and recovery Testing the on-line time response Testing the access to data Sequence testing Testing the complexity Testing the batch processing response</td>
</tr>
<tr>
<td>Nonfunctional requirement testing</td>
<td>Security testing</td>
<td></td>
</tr>
<tr>
<td>Front end testing</td>
<td>OLAP Cube testing</td>
<td></td>
</tr>
</tbody>
</table>

5.3 Best practices for Datawarehouse testing

Test Data - Datawarehouse testing is data intensive activity. Importance of testing with good amount of data has been stressed in literature. [21] and [19] have proposed generating test data by studying source data systems if sufficient test data is not available. [22] Recommends the testers at data source organizations to create data based on various scenarios while exploring the data source systems. This study also recommends carefully selecting databases of various sizes for different types of testing.

Importance of automation - Use of automated tools for testing has been proposed by majority of studies. Automating processes whenever possible will save tremendous amounts of time [24]. Automated testing of ETL has a positive impact on improving quality of data in Datawarehouse [18]. Early integration of testing into development efforts is possible. A process for automated test case and test data generation and testing of ETL processes has been advocated in [19].

Skills for DW testing - Skills required for DW testing team can be summarized as follows [26]:

67
5.4 Challenges in DW testing

A detailed insight into challenges faced during ETL testing categorized into 3 categories has been provided in [22].

VI. LIMITATIONS OF TESTING TECHNIQUES

Following is a summary of gaps that need to be addressed about Datawarehouse testing:

1. **Lack of standard test process** – Datawarehouse testing is a relatively new field compared to other testing fields and lack of standard testing process is a challenge with it. Lack of standard test process often leads to insufficient coverage, inconsistency and redundant test efforts [21]. In [16] an approach for standardization has been proposed, but it needs to be empirically and practically validated.

2. **Lesser focus on model driven and automation** – Datawarehouse testing literature has constantly referred to automation as a critical success factors. However, no extensive approach has been proposed for end to end automation of DW testing including generation of test cases, test data and result comparison.

3. **No studies about impact of DW testing on success of DW and quality of data** – Various papers have described challenges of DW testing. However, relationship between testing techniques and failures of DW, poor data quality in DW challenges has not been
explored. Such a study would help practitioners focus their attention on challenges of testing that cause highest rate of failure in DW systems.

4. **No study on skills used in the industry for DW industry** – [22] has stated lack of skilled resources as one of the challenges in DW testing. However, there has not been much research about skills possessed by DW testers in industry and areas of improvement. A research in this area will help human resource planning for DW projects.

5. **Data volume challenges** - DW testing requires testing with huge data. Ability to perform adequate testing with huge volume of data is a key to successful DW testing. Techniques for testing with huge volume have not been researched. Ability partition data and identify correct set of test data from production data needs to be researched. Ability to conduct testing in absence of sufficient data needs to be explored.

6. **Testing of Non functional requirements not given due attention** – Compared to testing of functional requirements for DW, testing of non-functional requirements has received minimal attention. Performance and usability of DW have been stated as important nonfunctional requirements that need to be tested [22]. However, in general testing of nonfunctional requirements needs to be given much more attention.

**VII. WORD CLOUD**

Word clouds are a visual depiction of the frequency tabulation of the words in any selected written material. Font size is used to indicate frequency, so the larger the font size, the more frequently a word is used [13]. In this paper, we have used word cloud as a technique to summarize key topics being discussed in the literature used as a reference for this paper. Detailed word clouds were drawn for each paper and for all papers together to understand topics that are discussed or ignored in literature. Following diagram shows key terms being discussed for references used in this paper:

![Word cloud for DW requirements literature](image1)
![Word cloud for DW testing literature](image2)

**Fig.2: Word cloud for requirement gathering and testing literature**
VIII. CONCLUSION AND DIRECTIONS FOR FUTURE RESEARCH

In this paper we studied various requirement gathering and testing techniques followed for datawarehousing. Merits and demerits of various traditional requirement gathering approaches like goal driven, user driven and source driven approach were studied. Lack of focus on non-functional requirements, need for CASE tools, lack of connectivity of requirement gathering techniques with design process were some of the drawbacks that were highlighted for traditional requirement gathering processes. Similarly, various aspects of DW testing were studied along with best practices proposed by different studies. Unavailability of tools for automation of DW testing, lack of standard testing process came out as important drawbacks of traditional DW testing processes. Empirical studies highlighting relation between various DW requirement gathering and testing techniques and success of Datawarehouse initiatives are absent in literature.

In future we plan to conduct empirical studies to connect success of DW initiatives with requirements and testing techniques. Such a research will enable practitioners to focus attention on avoidable causes of datawarehousing failure. We also plan to conduct a study of possibility of using emerging agile techniques for Datawarehouse development to overcome challenges mentioned in the paper.

IX. REFERENCES


