

Software Testing for BI

Lesson 4: Testing for BI

Lesson Objectives

- To understand the following topics:
 - Testing document purpose (Test documentation)
 - General BI Testing Principles
 - BI Testing Mission
 - Production Verification Testing
 - Possible Areas of Automation



4.1: Testing document purpose

Test documentation

- The purpose of this document is to define a template for the Testing Strategy of Business Intelligence Solutions.
- The document will:
 - Identify and describe each testing phase
 - Identify and describe each test type
 - Define entry and exit criteria of each test phase including test activities and validation tasks

4.1: Testing document purpose

Test documentation (Cont...)

- Each phase will have its own test plan prepared and accepted that will cover all aspects in detail appropriate to that phase.
- It will reference this document where necessary.
- The document also sets some high-level guidelines for BI testing.

4.2: BI testing principles

General BI testing strategy

- For each Reporting Solutions, all canned reports specified with the business should each be tested individually.
- To demonstrate the add-hoc capabilities of the system, the solution development vendor should demonstrate that all canned reports within the scope of the application can be produced using the ad-hoc toolset.
- Following release to production there should be process to support retro fitting of minor changes done by the application-support team and with major changes done by the development team.

4.2: BI testing principles

General BI testing strategy (Cont...)

- BICC should record defects in a tool (such as excel or better) with the intention to display bugs per modules and bug discovery rate
- As data quality is a major consideration in these type of projects the following should apply:
 - Data quality issues should be fixed in the upstream system, rather in the data warehouse.
 - A review of the data quality should be performed at each project decision point.

4.2: BI testing principles

General BI testing strategy (Cont...)

- The usability test should be conducted as follows.
 - Step 1 - Untrained users should be trained on the usage of the system.
 - Step 2 - These users should then be given a list of tasks to perform within the environment. The users performance should be monitored.
 - Step 3 – Based on the users feedback and performance either the training data set, or the training material should be improved.

4.2: BI testing principles

General BI testing strategy (Cont...)

- In case of unavailability of sufficient test data a test data-generating engine such as GS Data Generator or DBUnit may be used.
- For data security purposes, test data should be generated by appropriately massaging production data

4.3: iGATE BI testing mission

BI testing mission

- BI Testing Mission

- “The objective of testing is to verify that the deliverables of BI solutions meets the agreed business requirements, and comply with both IT and BI operations as well as domain standards.”

4.4: Production verification testing

Production Verification Testing

■ Production Verification Testing (PVT)

- It verifies that the correct functionality, architecture and procedures have been defined and implemented.
- It allows for the running, maintenance and support of the system in production in accordance with common practice and agreed standards.
- PVT also confirms that all necessary hardware, software, application functionality migrated from validated test environments, converted data, connectivity and interface capability are available and confirmed for the system to be implemented into production.

4.5: Possible areas of automation

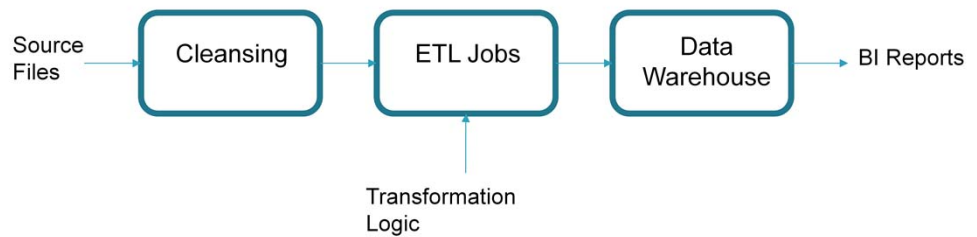
Possible areas of automation

ETL Layer	<ul style="list-style-type: none">▪ Generic Data Reconciliation: Programs to automate Data Reconciliation process▪ Generic Data Auditing Routines: Programs to collect statistics related to the data being loaded from source to target
Portal/OLAP Reporting Layer	<ul style="list-style-type: none">▪ Portal Look and Feel Check▪ Reports Generation & Data Validation (CSV File Compare)
Performance /Stress Testing	<ul style="list-style-type: none">▪ ETL Process Performance▪ Report Generation Time▪ Response times OLAP & Presentation Layer

ETL Testing

- ETL Testing :

- Cleansed Data from the Source file is loaded into Data Warehouse after implementing the transformation logic
- Source Files can be any of Flat Files, XML Files or Database Tables.



Continuation.....

- Validating the Transformation logic implemented in ETL Jobs while loading the Data from individual Source system to Data Warehouse.
- Compare the data in Data Warehouse with source data according to the logic given in mapping document.
- Mapping Document contains the Information regarding..
- Source System :Source File Name, Source Field(s) and its Data type
 - Transformation Logic.
 - Target System (Data Warehouse) :Target Table Name , Target Column Name and its data type.

ETL Testing Using SQL

- Source Data Tables: Data from source files is replicated into data base tables using any SQL Loader or Simple ETL Jobs.
- Using SQL, Target(DWH) and Source tables are compared with Minus operator.
- Applying the transformation logic on source fields according to the logic given and comparing it with the field in target.
- Syntax :
 Select T/L (Source_Field) from Source_Table
 Minus
 Select Target_Field from Target_Table
- No Logic to be implemented on Target.

Source System Unique Key

- All Attributes in the Target Table are validated with reference to Unique Key in the Table.
- For Each Table in the Data Warehouse Unique key is implemented based on logic given in Mapping Document.

T/L(Src_Fld1,Src_Fld2)--> Unique_Key_DW

Syntax:

Select Uniq_Ky, Source_Field from source_Table

Minus

Select Uniq_Ky,Target_Field from Target Table

Smoke Test

- It confirms that total no. of records present in both Source and Target are equal.
- It's a basic test for testing a Data load/Migration.
- Syntax:

Select count (distinct Uniq_Ky) from Source_Table.
Select count (distinct Uniq_Ky) from Target_Table.

SSUK Check

- Assuring that all the records present in Source Table are loaded into the Target Table.
- Syntax:
 - Select distinct Uniq_Ky from Source_Table
 - Minus
 - Select distinct Uniq_Ky from Target_Table

Attribute Testing

- Validating the Transformation Logic for all attributes with reference to the Unique_Key.
- Syntax:
Select Uniq_Ky, T/L(Source_Field) from source_Table
Minus
Select Uniq_Ky, Target_Field from Target Table
- Some of the fields in DWH are set to a default value say '0/NULL/Space'
- Validating the Default Set Fields (0/NULL/Space) in DWH:
Select Count(*) from Target_Table
Where Target_Field <> '0/Null/Space'

Multiple Loads

- Data gets loaded into Data warehouse usually in different loads for various Periods, say monthly, Bi-Monthly, Quarterly, Half yearly or Yearly based on Business Needs.
- Full File Load : From Second Period Onwards Snapshot of Total Records at second period appear in Source File.
- Delta File Load: From Second Period Onwards Only Changed Records appear in Source File.

CDC

- CDC : Change Data Capture in Data Warehouse.
- Data from Multiple Loads is controlled in DW using Two Date Fields in the Table Say EFFV_DATE and END_DATE.
- For an Active record EFFV_DATE is the Period Load Date and END_DATE is any Unused Long Date.
- For example when the first load is on 01-May-2011, the EFFV_DATE will be '01-May-2011' and END_DATE will be '31-Dec-9999'

Continuation.....

- When the second load occur on '01-Jun-2011', for a given Uniq_Ky,
- 1) When the records for both Periods Matches, then no New Record is Inserted for P2 in DW and No change in Action Flag.
- 2) When the record in the Period 1 is deleted in Period 2, then END_DATE for Period 1 record gets changed to '31-May-2011' in DW.
- 3) When the record in Period 2 is a newly inserted record and not exist in Period 1, then a new record inserted with EFFV_DATE as '01-Jun-2011' and END_DATE as '31-Dec-9999'
- 4) When the records for both periods not matches, then a new record inserted with EFFV_DATE as '01-Jun-2011' and END_DATE as '31-Dec-9999' and the END_DATE for existing record with EFFV_DATE as '01-May-2011' gets changed to '31-May-2011'.

Continuation.....

- Example:
- Period 1 Load Date : '01-May-2011
- Period 2 Load Date : '01-Jun-2011

Period	Unchanged	Deleted	Inserted	Changed
Period 1	EFFV_DT: '01-May-2011' END_DT : '31-Dec-9999'	EFFV_DT: '01-May-2011' END_DT : '31-May-2011'	No Records	EFFV_DT: '01-May-2011' END_DT : '31-May-2011'
Period 2	No Records	No Records	EFFV_DT: '01-Jun-2011' END_DT : '31-Dec-9999'	EFFV_DT: '01-Jun-2011' END_DT : '31-Dec-9999'

Modified Smoke and Attribute Test Scripts:

- In the Target Side condition for restricting the records for that specific period is included.
- SSUKT Check:
Select distinct Uniq_Ky from Source_Table_P1
Minus
Select distinct Uniq_Ky from Target_Table
Where Load_Date between EFFV_DATE and END_DATE
- Attribute Validating Script:
Select Uniq_Ky, T/L(Source_Field) from source_Table_P1
Minus
Select Uniq_Ky,Target_Field from Target Table
Where Load_Date between EFFV_DATE and END_DATE

CDC Check

- For Full File:

- 1) Matched record Check.

- Source Query:

- Select Uniq_Ky, T/L(Field 1),T/L(Field 2) from Source_Table_Period_1

- Intersect

- Select Uniq_Ky, T/L(Field 1),T/L(Field 2) from Source_Table_Period_2

- Target Query:

- Select Uniq_Ky, Effv_dt,End_Dt from Target_Table

- Where Uniq_Ky in(Sample records from Source Query Result)

- Expected Result: There should be one record for each Unique Key with
EFFV_DT as '01'May-2011'

Continuation.....

2) Deleted Record check:

Source Query:

```
Select Uniq_Ky from SRC_P1  
Where Uniq_Ky not in (Select Uniq_ky from SRC_P2)
```

Target Query:

```
Select Uniq_Ky,EFFV_DATE,END_DATE from Target_Table  
Where Uniq_Ky in(Sample records from Source Query Result)  
Only one record for each Uniq_Ky with EFFV_DATE '01-May-2011' and  
End Date as '30-Jun-2011'
```

Continuation.....

3) Inserted Record s Check:

Source Query:

```
Select Uniq_Ky from SRC_P2  
Where Uniq_Ky not in (Select Uniq_ky from SRC_P1)
```

Target Query:

```
Select Uniq_Ky,EFFV_DATE,END_DATE from Target_Table  
Where Uniq_Ky in(Sample records from Source Query Result)  
Expected Result: Only one record for each Unique Key with EFFV_DATE as '01-Jun-2011' and End Date '31-Dec-9999'.
```

Continuation.....

4)Un-Matched Record Check.

Source Query:

Select Uniq_Ky,T/L(Field 1),T/L(Field 2) from Source_Table_Period_1

Minus

Select Uniq_Ky,T/L(Field 1),T/L(Field 2) from Source_Table_Period_2

Target Query:

Select Uniq_Ky,EFFV_DATE,END_DATE from Target_Table

Where Uniq_Ky in(Sample records from Source Query Result)

Expected Result: There should be Two record one with EFFV_DATE as '01-May-2011' and END_DATE as '31-May-2011' and the other with EFFV_DT as '01-Jun-2011' and END_DATE as '31-Dec-9999'.

But the Source Query gives both Un-Matched and Deleted Records.

Continuation.....

- Modified Source Query for Unmatched Record Check:
- Select Uniq_Ky,T/L(Field 1),T/L(Field 2) From Source_Table_Period_1
- Where Uniq_Ky in(Select Uniq_Ky From Source_Table_Period_2) Minus
- Select Uniq_Ky,T/L(Field 1),T/L(Field 2) From Source_Table_Period_2

- Suggested Query To Overcome Performance Issue:
 - Select Uniq_Ky From Source_Table_Period_1 P1
 - Inner join Source_Table_Period_2 P2
 - On(P1.Uniq_Ky=P2.Uniq_Ky and P1.Fld_1 <> P1.Fld_1 and P1.Fld_2<> P2.Fld_2)

Continuation.....

- For Delta File:
 - For Unchanged Records: No Record Present in Period 2 Load.
 - Select Uniq_Ky from SRC_P1
- Where Uniq_Ky not in (Select Uniq_ky from SRC_P2)
- For Changed Records : An action field exist.
 - Select Uniq_Ky from Source_Table_Period_2 where action = 'Insert/Delete/Modified'

Duplicate Record Check In CDC

- `Select uniq_Ky,EFFV_DT_END_DT ,count(*)from target_Table
group by uniq_Ky,EFFV_DT_END_DT having count(*) > 1`

DATA COMPLETENESS AND QUALITY CHECK

- An integral part of DWH testing is verifying the quality and completeness of data. Data completeness testing ensures that all expected records from the source are loaded into the database by reconciling with error and reject records. A data quality check ascertains proper and accurate data, as per the recommended standard, is processed to the data warehouse; this includes data transformation testing.

DATA COMPLETENESS AND QUALITY CHECK

- The following activities are recommended, to determine data completeness and quality:
- Data extraction process for both historical and incremental loads
- Data cleansing checks, based on standards; here the testing reject threshold is important
- 'Source to target' transformation validation for thoroughness and accuracy
- Historical and incremental transformation process validation
- 'Reject and error' record analysis and validation
- Scenario-based testing with specified transformation rules
- Record reconciliation testing by comparing source, error, reject and target records to prevent record leakage
- Data load process check for both historical and incremental load process
- Negative testing for all the above mentioned cases

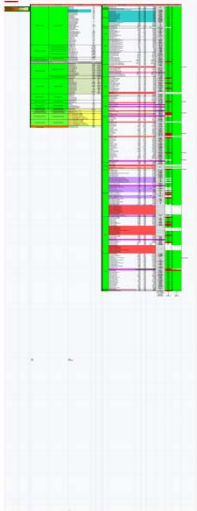
What Are Key Performance Indicators (KPI)


- Key Performance Indicators (KPI) are used by organizations to evaluate success and when you choose KPI's you should follow the smart approach
- Specific – a well defined goal that is clearly understood by everyone.
- Measurable – can you track your progress towards the goal?
- Agreed – both employer and employee must agree on what the goals are.
- Realistic – can you achieve the goal with the resources provided?
- Time related – will there be enough time to complete the task?

So what is a “Report”?

- A report is made up of three components:
 - Data: specifies how to extract information from backend data sources and information on the structure of that data.
 - Layout: how the information is to be presented.
 - Properties: parameters, interactions, etc.
- Typically, the report is re-used at intervals.
 - It picks up the current data from the data sources.
- The report definition may be stored in XML.
 - An XML report template can be used to define a family of related reports

KPI Testing



Copyright © Capgemini 2015. All Rights Reserved 35

BI REPORT DATA TESTING

- Another important aspect of DWH testing is confirmation of the accuracy and completeness of business intelligence (BI) reports. These may vary in appearance, turnaround time, report accuracy and usability, but testing this is of paramount importance as this will be reflected in the UI and is what the end users will eventually see. The following activities are key while testing BI reports:
 - Restriction of users' access to reports, with multiple layers of security
 - Validation of the accuracy and relevance of the data displayed in each report
 - Ensuring sufficient information for analyzing graphical reports
 - Relevancy of options in the drop down lists in each report
 - Testing of pop-up reports and child reports with proper data flow from parent reports
 - Functionality of additional features such as report storage into PDF formats, print options

Summary

- In this lesson, you have learnt:
 - Test document identifies and describes each testing phase.
 - Each phase has its own test plan prepared and accepted.
 - The objective of testing is to verify that the deliverables of BI solutions meets the agreed business requirements.
 - contd.



Summary (Cont...)

- In this lesson, you have learnt:
 - Production verification testing confirms the system to be implemented into production.
 - There are various areas of BI where in testing can be automated.



Review Questions

- Question 1: _____ is to define a template for the Testing Strategy of Business Intelligence Solutions
- Question 2: Data quality issues should be fixed in the Data warehouse
 - True / False

