<epam>

Data Quality

Introduction to DWBI Systems Testing

TRAINING

CENTER

<epam>

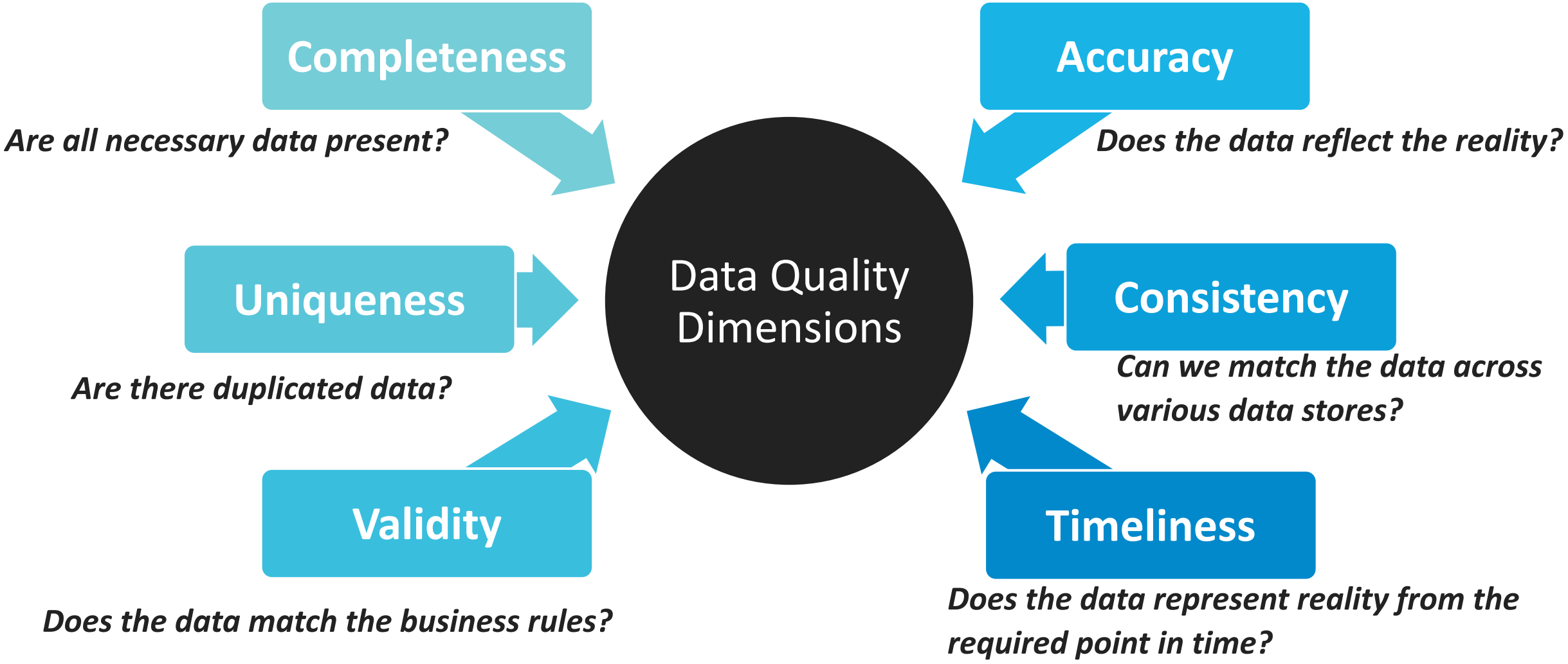
<epam>

Agenda

- Data Quality Dimensions
- DWBI Testing Types
- DWBI Testing Levels

DATA QUALITY DIMENSIONS

DATA QUALITY DIMENSIONS



DATA QUALITY DIMENSIONS

Measure	Description	Examples / Checks
Completeness	A measurement of the availability of required data attributes. Measure of the amount of data provided in proportion to the amount of data possible.	• By records/columns/triggers/functions • By empty and non-empty fields • Get sums of numeric, decimal, integer columns • Get sum of date and datetime columns
Uniqueness	A measurement of the degree that no record or attribute is recorded more than once. Refers to the singularity of records and or attributes.	• Check for record duplicates / files duplicates • Check that primary keys are developed and there are no duplicates
Accuracy	Measurement of the precision of data. Shows how closely data represent the state of the real world at any given time. It can be measured against either original documents or authoritative sources and validated against defined business rules.	• Records that are wrong at a specified time • Records that haven't been refreshed or updated • no negative values where not expected • no future dates where not expected • verify date-time conversion across the all data • any 'x' value equals to 'x' that is considered to be correct
Validity	Refers to information that doesn't conform to a specific format or doesn't follow business rules	• data type / data length /min, max values/range/allowed values • check precision of each numeric field • check default values

DATA QUALITY DIMENSIONS

Measure	Description	Examples / Checks
Consistency	The absence of difference, when comparing two or more representations of a thing against a definition. Refers to data values in one data set being consistent with values in another data set. A strict definition of consistency specifies that two data values drawn from separate data sets must not conflict with each other Types: 1. Record-level consistency 2. Cross-record consistency 3. Temporal consistency 4. Across different lines of business or apps	• Not logical given parameters or rules • Invalid data formats in some records in a feed • Telephone numbers with commas vs. Hyphens • Auto computed values based on other attributes
Timeliness	Refers to the time expectation for accessibility and availability of information. Timeliness can be measured as the time between when information is expected and when it is readily available for use	• A file delivered too late for a business process or operation • A credit rating change not updated on the day if was issued

More on the topic: https://silo.tips/download/the-six-primary-dimensions-for-data-quality-assessment

DWBI TESTING TYPES

Functional

- **Data Completeness testing** – ensure that all expected data is loaded by means of each ETL procedure
- **Data Transformations testing** – ensure that all data is transformed correctly according to business rules and/or design specifications
- **Data Quality testing** - ensure that the ETL process correctly rejects, substitutes default values, corrects, ignores, and reports invalid data

Non-Functiona

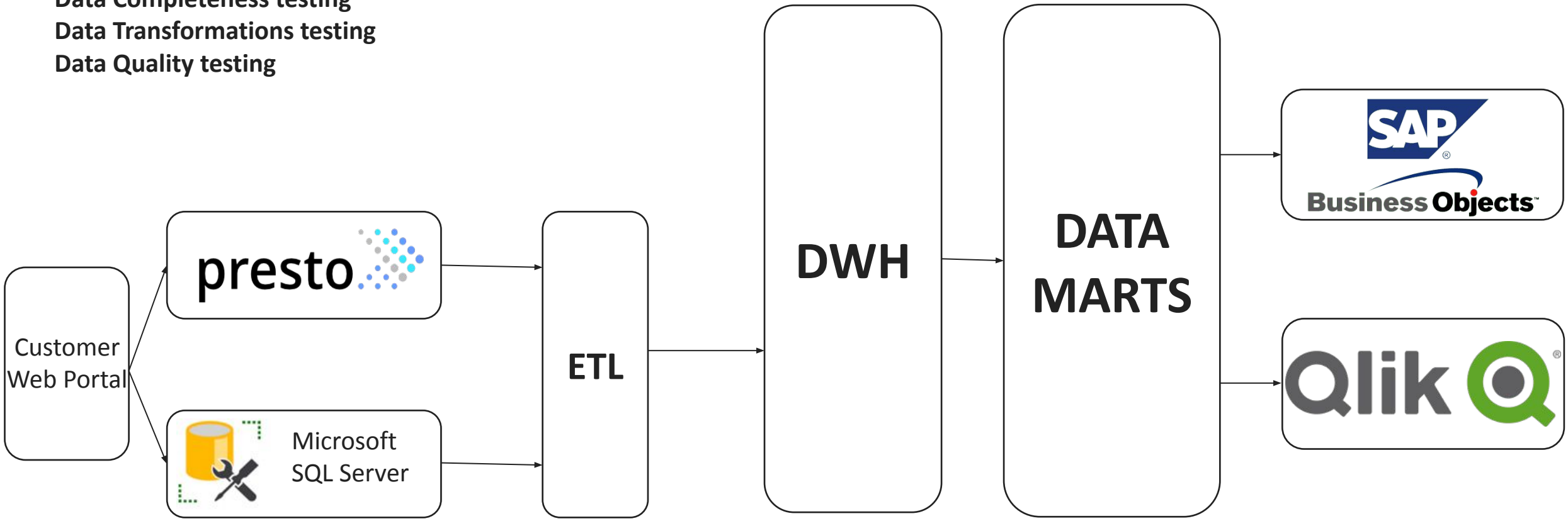
- **Performance and scalability testing** – ensure that data loads and queries performs within expected time frames and that the technical architecture is scalable
- **Security testing**.

Regression

Testing activities to ensure existing functionality remains intact each time a new release of ETL code and data is completed.

SYSTEM SCHEMA

Data Completeness testing
Data Transformations testing
Data Quality testing

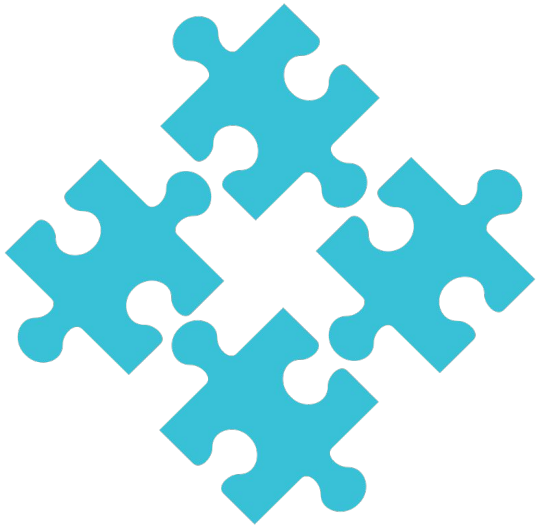


REGRESSION

DATA COMPLETENESS TESTING

The purpose of Data Completeness testing is to verify that all the expected data is loaded to target from the source.

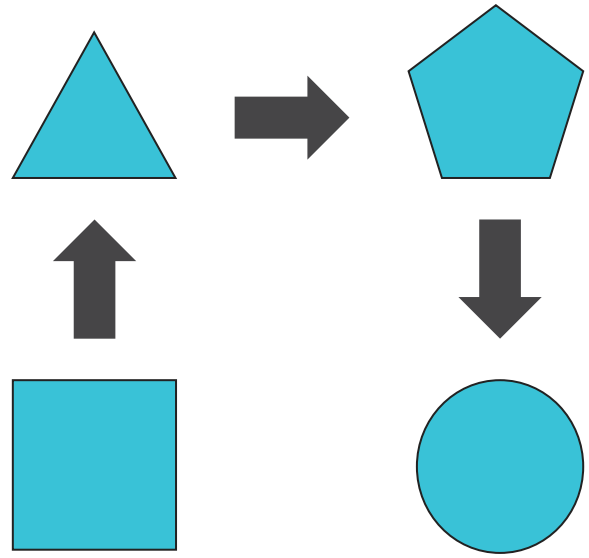
One of the most basic checks will be to verify that all records, all fields and full content of each field are loaded. Some of the tests that can be run are: Compare and Validate counts, aggregates (min, max, sum, avg) and actual data between the source and target.



Data transformations testing – ensure that all data is transformed correctly according to business rules and/or design specifications.

One of the most basic checks will be to validate that data is transformed correctly based on business rules. Steps to perform:

1. Review the source to target mapping / requirements to understand the transformation logic.
2. Apply transformations on the data using SQL or generate expected result for test data.
3. Compare the results of the transformed test data with the data in the target table.



Data quality testing - ensure that the ETL process correctly rejects, substitutes default values, corrects, ignores, and reports invalid data.

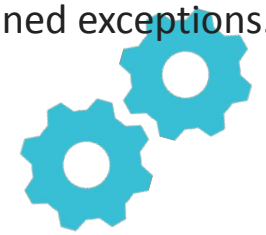
ETL is designed to fix or handle data quality issues . However, source data keeps changing and new data quality issues may be discovered even after the ETL is being used in production.

- Duplicate Data Checks
- Data Validation Rules
- Data Integrity Checks

Functional Testing – ensures that the application functions in accordance with design specification.

It considers the following verifications:

- Verify data mappings, source to target;
- Verify that all tables and specified fields were loaded from source to staging;
- Verify that primary and foreign keys were properly generated using sequence generator or similar;
- Verify that not-null fields were populated;
- No data truncation in each field;
- Data types and formats are as specified in design phase;
- No unexpected duplicate tables records in target tables;
- All cleansing, transformation, error and exception handling;
- Stored procedure calculations and data mappings
- Verify transformation based on data table low level design
- Numeric fields are populated with correct precision;
- Each ETL session completed with only planned exceptions.



NON-FUNCTIONAL TESTING - PERFORMANCE

Performance testing checks the speed, response time, reliability, resource usage, scalability of a software program under their expected workload. The purpose of Performance Testing is not to find functional defects but to eliminate performance bottlenecks in the software or device.

DQE checks:

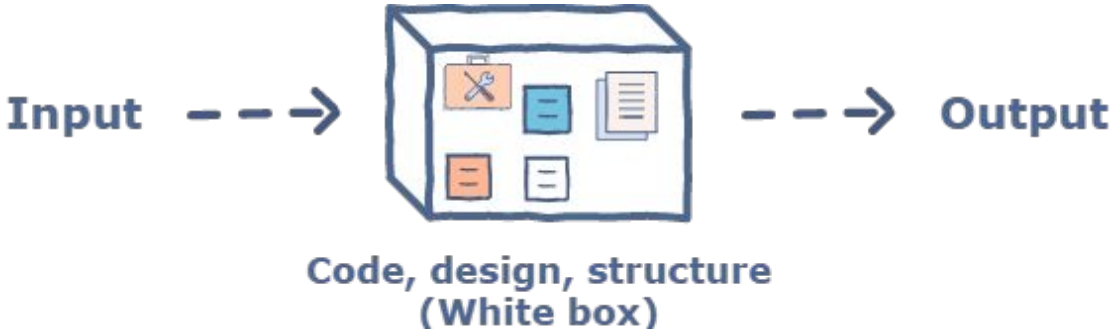
1. Perform ETL with peak expected production volumes to ensure it finishes within the agreed-upon time window;
2. Compare peak ETL loading times to ETL performed with a smaller amount of data;
3. Track ETL processing times for each component to point out any bottlenecks;
4. Track the timing of the reject process and consider how large volumes of rejected data would affect the performance;
5. Perform simple and multiple join queries to validate query performance on large database volumes. Work with business users to develop sample queries and acceptable performance criteria for each query.

Security Testing

The aim of the security testing is to check:

- The correctness of role-based access.
- The correct work of data encryption and decryption implementations adopted by the business.
- The validity of data backup operations.
- Query generation

BLACK & WHITE BOX TESTING



Black-box testing examines the functionality of an application without looking at internal structures for transformation testing.

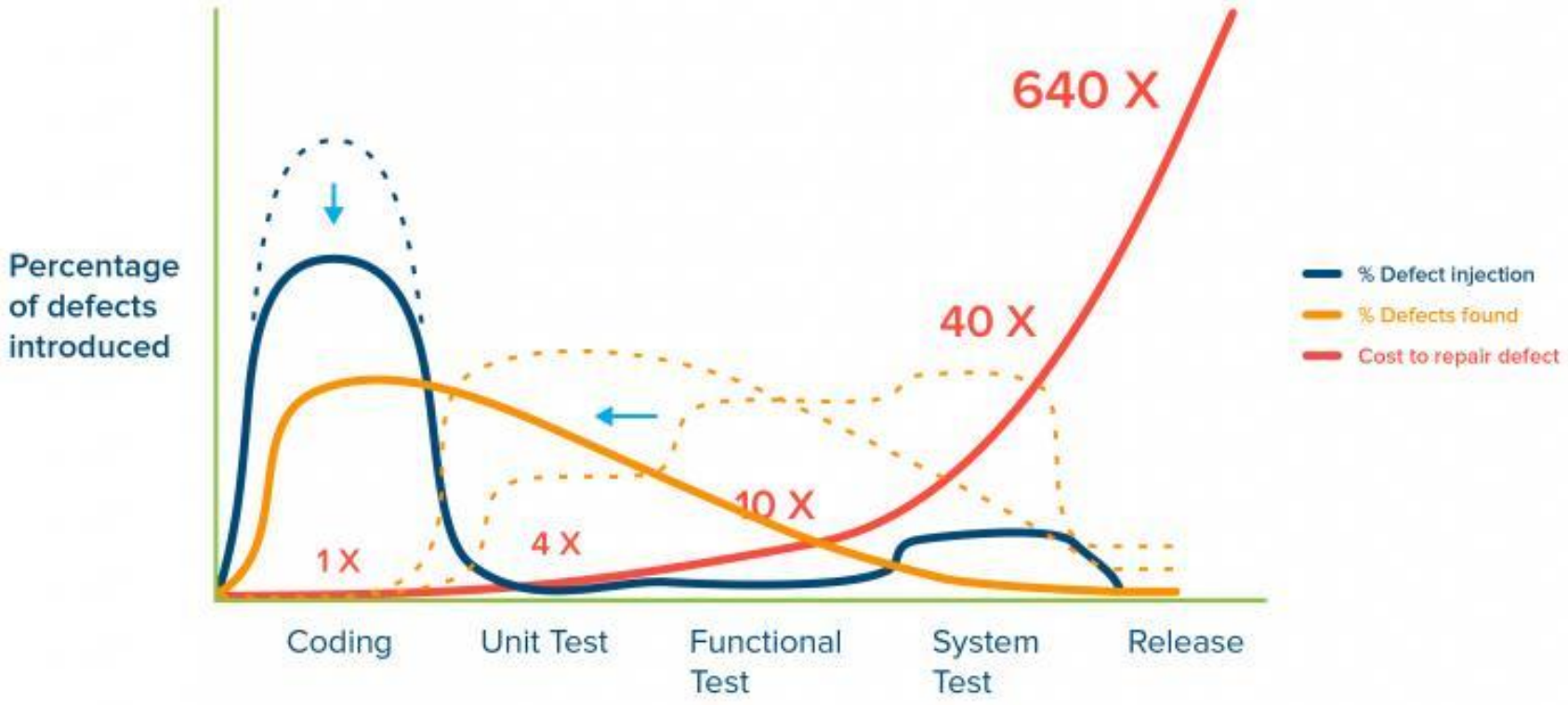
Testers review the transformation logic from the **mapping design document** creating the appropriate test data.

White box data transformation testing examines the program structure and develops test data from the program logic/code.

Testers review the transformation logic from the **mapping design document and the ETL code** to create test cases.

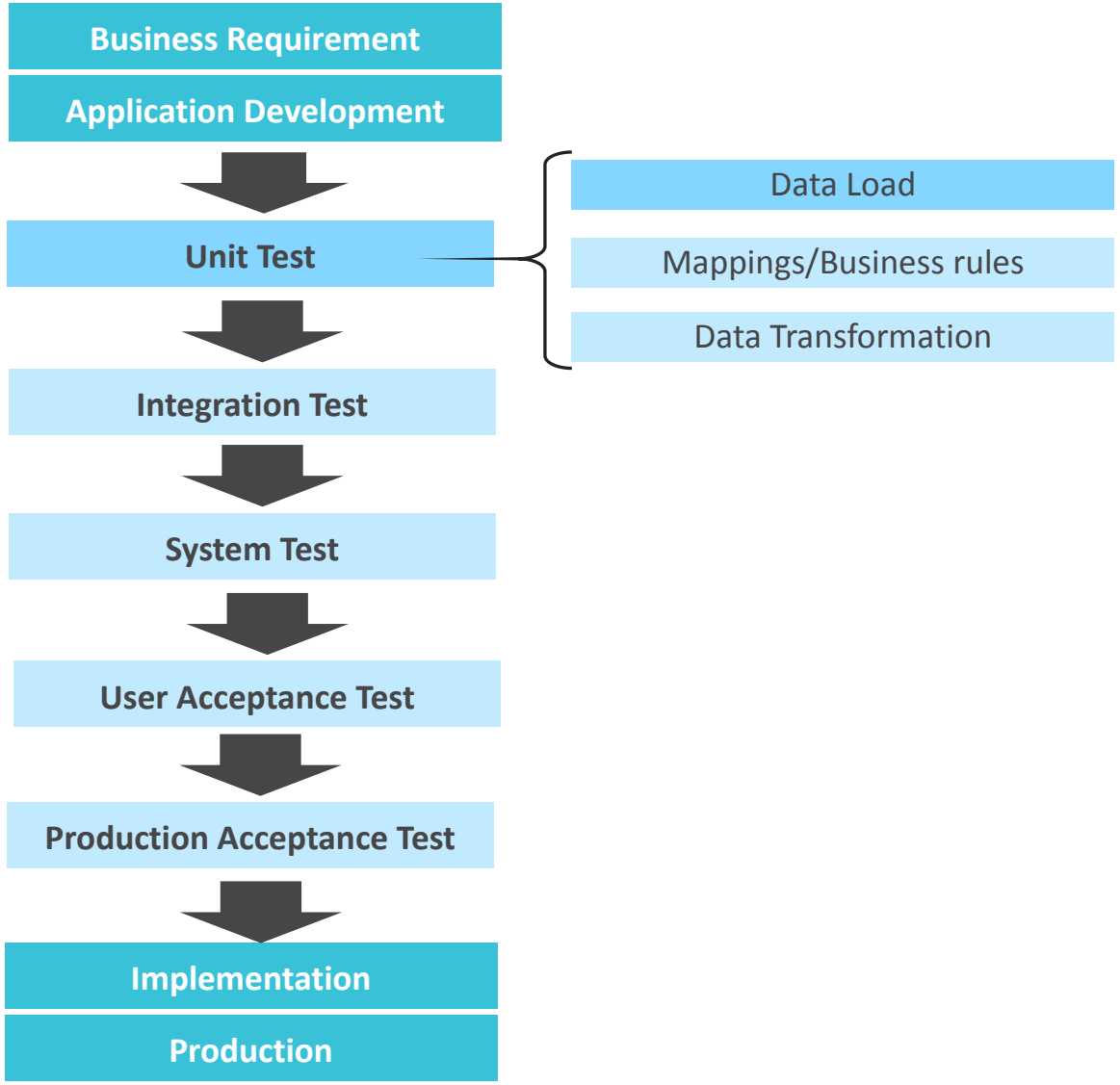
DWBI TESTING LEVELS

COST OF DEFECT



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*

TEST LEVELS


Business Requirement
Application Development
Unit Test
Data Load
Mappings/Business rules
Data Transformation
Integration Test
System Test
User Acceptance Test
Production Acceptance Test
Implementation
Production

TEST LEVELS

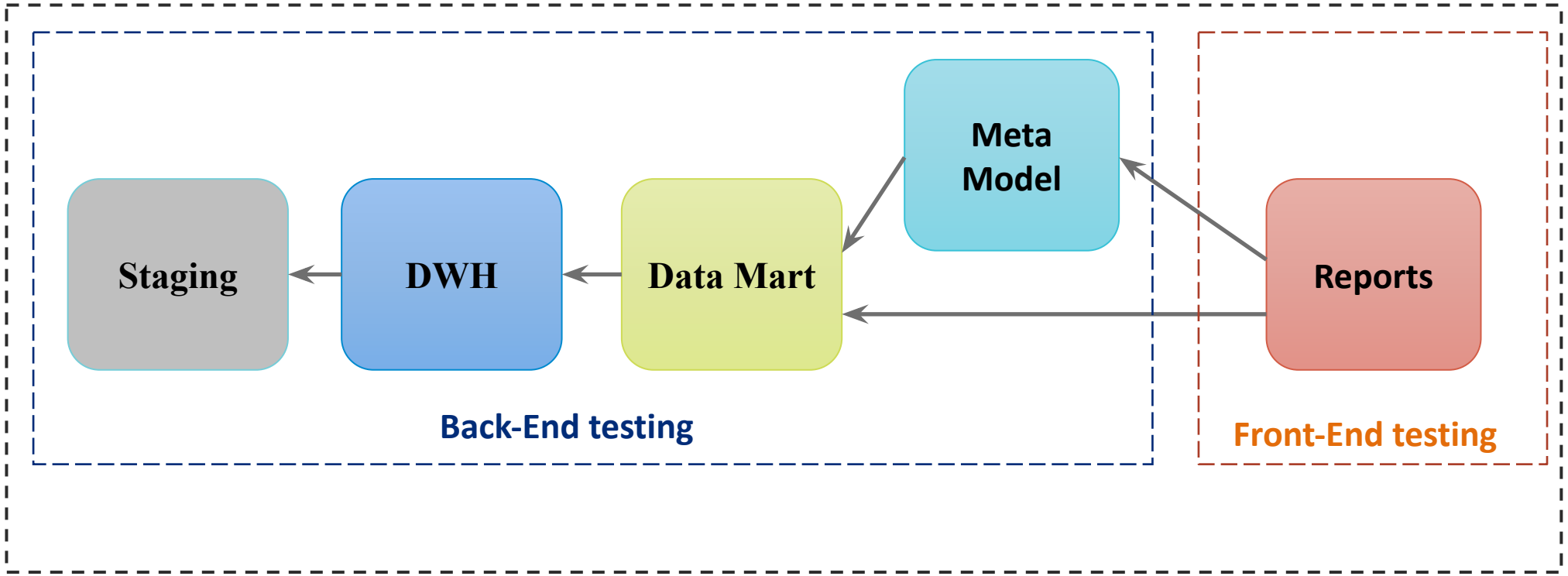
- **Component (Unit) Testing** – is done at the lowest level. It tests the basic unit of software, which is the smallest testable piece of software, and is often called unit, module or component.

- **Integration Testing** – ensure that the ETL process functions well with other upstream and downstream processes.

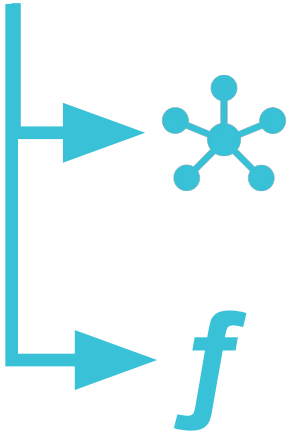
- **System Testing** – ensures that the ETL application functions well.

- **User-acceptance Testing** – ensure the data warehousing solution meets users' current expectations and anticipates their future expectations.

DW/BI unit testing is divided into two parts



Back-end testing can be divided by:

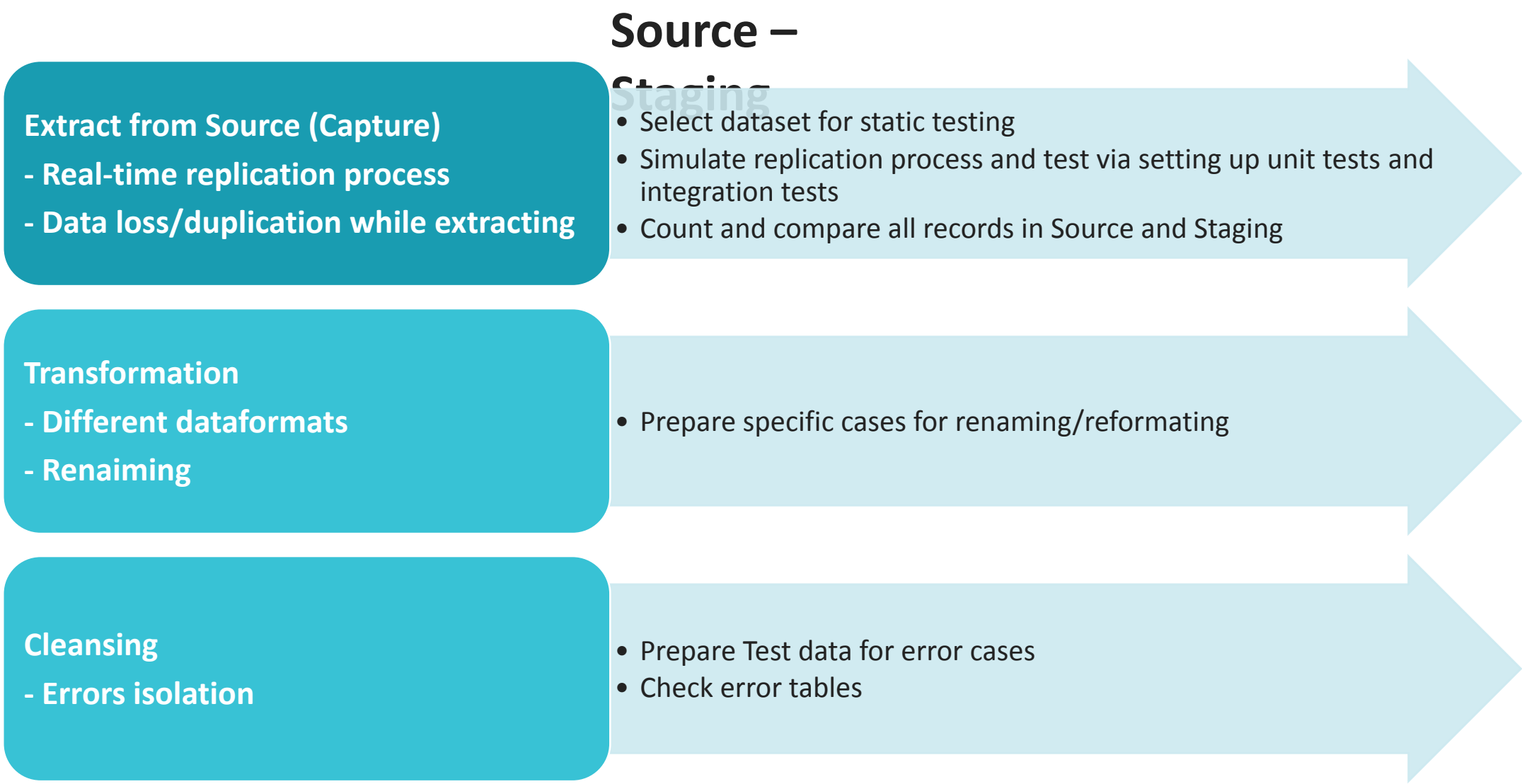


dimensions/facts

f functions such as extraction, incremental extraction, cleaning, transformation, static loading, incremental loading

Back-end testing can be divided into few steps. In each step the specific processes should be tested.

UNIT TESTING – BACK-END



Staging – DWH

Extract from Staging (Capture)
- Incremental and Full load
- Errors isolation

- Simulate CDC (Change Data Capture) process
- Check error tables and rules

Consolidation
- correct business rules for data cleansing, consolidation and merging
- Missed/corrupted records

- Count and compare all records in Staging and Store and test via setting up unit tests and integration tests

DWH - Data Marts

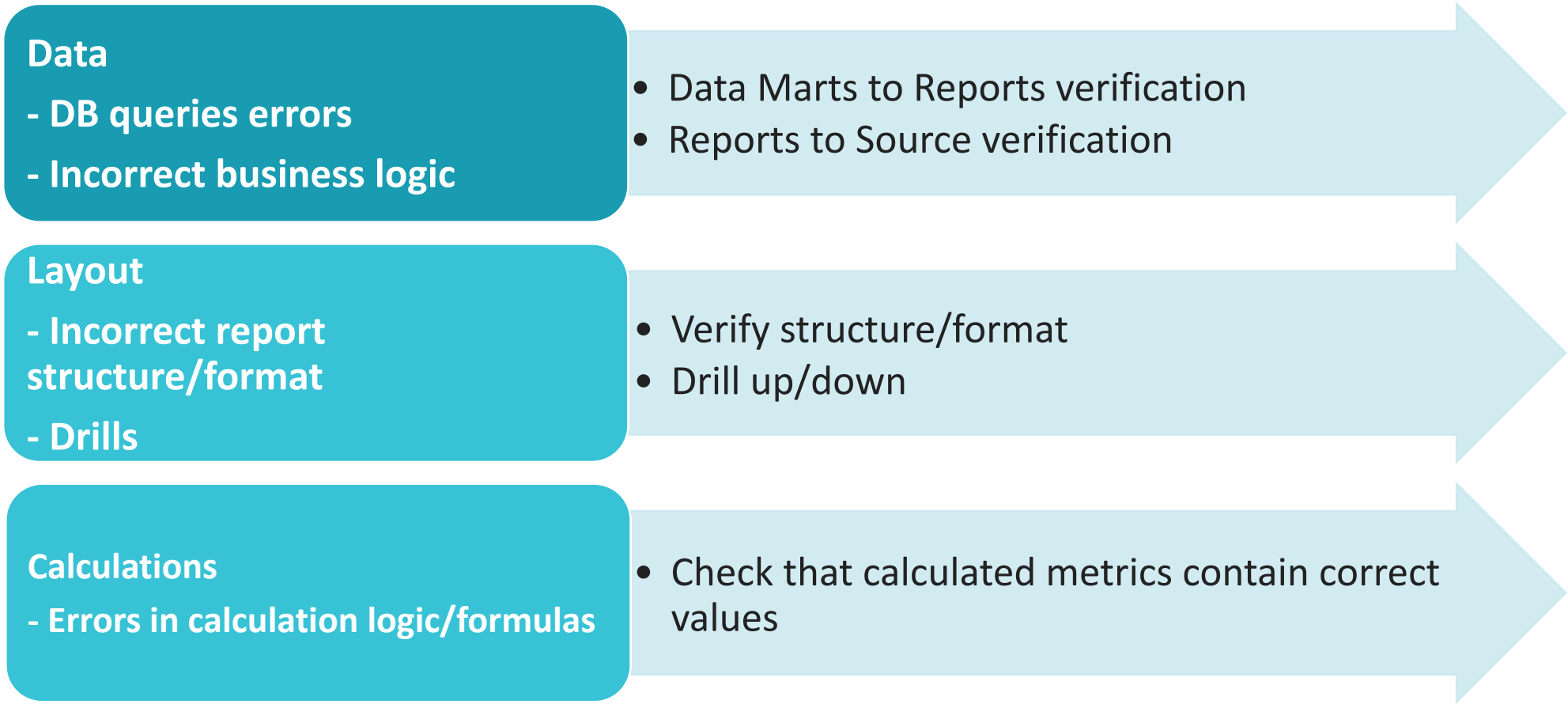
Aggregations
- Aggregation functions errors
- Data Filters

- Verify that aggregate functions are performed against expected data set and in proper way
- All invalid data is filtered out

Calculations
- Calculation logic errors
- Incorrect formulas

- Test Calculations
- Metrics and dimensions verification

UNIT TESTING – FRONT-END



Integration testing shows how the application fits into the overall flow of all upstream and downstream applications

Focus on touch points between applications rather than within one application

SYSTEM TESTING

ETL testing using black-box test methodology based on design document.

The whole data warehouse application is tested together.

The purpose of system testing is to check whether the entire system works correctly together or not.

TEST LEVELS EXAMPLES

Unit tests

- Procedure correctly loads data from source to target table (1 to 1 or with transformations);
- E-mail sent to a team if a procedure failed;
- Default value of dimensionID is inserted into fact if there is no corresponding record in the fact;
- Anonymization case: user email and phone replaced with aaa@test.com and 000.

Integration tests

- After adding new package to ETL job previously implemented functions are working okay;
- When new grouping is implemented in DM table dashboards are working fine;
- Some data from UI are saved in table. After changing table metadata data from UI correctly saved into database table.

System Test

- All features and functions of system are working correctly. All. UI, DB, ETL, Dashboards, etc.

UAT TESTING

User-acceptance testing typically focuses on data loaded into the DWH and any views that have been created on top of tables, not the mechanics of how the ETL application works.

Consider the following strategies:

- Use data that is either from Production or as near to production data as possible. Users typically find issues once they see the real data, sometimes leading to design changes

- Test DB views comparing view contents to what is expected. It is important that users sign off and clearly understand how the views are created

- The QA test team must support users during UAT. The end users will likely have questions about how the data is populated and need to understand details of how the ETL works

- During UAT, data will need to be loaded and refreshed a few times.

Acceptance Testing – is testing from the end users perspective, typically end-to-end, to verify the operability of every feature

Purpose

Rather to give confidence that the system is working than to find errors.

This is the most critical part of the test cycle because the actual users are the best judges to ensure that the application works as expected by them.

<epam>

Q & A