DATA QUALITY STRATEGY

POWER BI DASHBOARDS

Content

1. GENERAL INFORMATION

Document goal and scope: The document describes strategy and techniques for testing Power BI Dashboards:

- Power BI dashboard “Sales”
- Power BI dashboard “Cost”
- Power BI dashboard “Stocks”

It provides insight into what activities and steps within the activities are needed to ensure that the solution provides end users with the correct data.

Audience: The document is to be used by Quality Assurance team (QA team) and Project team as a guideline:

- of what needs to be tested
- how it needs to be tested
- what tools are to be used
- what acceptance criteria are to be used, etc.

Terms: Hereinafter, the Power BI dashboards project is called “**PBI Dashboard**” or “**PBID**”

2. ENTRY CRITERIA

Before starting the testing of a PBI Dashboard, there are several entry criteria that need to be met to ensure that the testing is effective and efficient. Some of them are:

- **requirements** for the PBI Dashboard should be clearly defined, documented, and approved by the stakeholders and provided to the QA team
- **data sources** for the dashboard (EDW in current project) should be identified and available for testing. The data sources should be complete, accurate successfully pass all necessary tests
- **design of the dashboards** should be finalized and approved by the stakeholders. The design should include all the required visualizations, filters, and drill-down functionality
- **test environment** should be set up with the required hardware, software, and data sources. The test environment should be representative of the production environment
- **test data** must be prepared and loaded into or accessible from the test environment. The test data should be representative of the production data and cover all the scenarios to be tested
- **test plan** should be created that includes the objectives, scope, testing approach, testing environment, and test cases. The test plan should also include the timeline and budget for testing. The test plan must be aligned with this test strategy
- **test cases** should be developed based on the requirements and design of the dashboards. The test cases should cover all the critical functionalities of the dashboards

3. POSSIBLE RISKS

There are some risks that can occur during the testing of a Power BI dashboard. These risks can impact the quality of the dashboard and delay the release. Some of the possible risks are:

- **data quality issues**. The data used in the dashboard may have quality issues such as inconsistencies, inaccuracies, or missing data. This can impact the accuracy of the visualizations and analysis
- **integration issues**. The dashboard may integrate with other systems, such as databases or APIs, which may cause integration issues, such as data inconsistencies or errors
- **security issues.** The dashboard may have security vulnerabilities, such as weak authentication or authorization, which can result in unauthorized access to data or a breach of sensitive information
- **compatibility issues.** The dashboard may not be compatible with all web browsers or operating systems, which can limit the dashboard's accessibility to users
- **scope creep.** The scope of the dashboard may change during the testing phase, which can lead to delays in testing and release
- **internal risks**. Internal risks may include lack of resources such as time, budget, skilled personnel; poor communication between the project team members, stakeholders, and end-users

To mitigate these risks, it is important to have a well-defined test plan, a clear understanding of the requirements and objectives of the dashboard, and thorough testing of all critical functionalities. Close collaboration between developers, testers, and stakeholders can also help to identify and mitigate risks during testing.

4. TEST APPROACH and ACCEPTANCE CRITERIA

The following testing techniques can be used in each stage of the test approach:

4.1. Functional testing

Functional testing should be performed to ensure that all the functionalities of the dashboard are working as expected.

Test Case Techniques:

- compare data between EDW and PBI Dashboard using test SQL scripts (metadata, calculations etc);
- verify that all needed filter values available in each filter/prompt;
- compare data between LND and EDW using SQL queries or requests to the Team2 and/or Team1 in case of any data discrepancies will be found, etc;
- compare all aggregations such as amount per month, amount per year, cumulative amount, total accumulated, sum per period etc. between EDW and PBI Dashboard;
- check data objects for duplicates, check null values handling;

- verify data for data types;
- verify that all functionalities and report options such as prompts, filters, drilling, sorting options, data and time and so on are accessible and work as required;
- verification of the summarized report options (buttons/controls for the possibility to sum, count, avg., total) ;
- verify that links/buttons from detail report to chart and vice versa are accessible and work as required;
- verify export functionality etc.;
- check the whole report displaying in required browsers and screen resolutions;
- check standard colors, fonts, header and footer;
- check graphs, grids, prompts, filters, widgets layout on the page, etc.;
- join each separate measure and all possible attributes from different dimensions and apply only required filter(s) (if any) to get the most extensive dataset for the comparison with SQL query results;
- join all measures related to the particular report area and all possible attributes from different dimensions and apply only required filter(s) (if any) to get the most extensive dataset for the comparison with SQL query results;
- for each dimension folder, pick all attributes and run report to verify that any attribute is not mapped properly;
- check the dimension hierarchy.

Quality and Acceptance Criteria:

- results of data comparison between PBI Dashboard and results of SQL scripts execution written against EDW level by the defined requirements should show no discrepancies between them;
- results of data aggregation comparison between PBI Dashboard and results of SQL scripts execution written against EDW level by the defined requirements should show no discrepancies between them;
- data is considered to have sufficient quality, if within expected boundaries, had expected accuracy and data type;
- all functionalities of each separate report should work as it was described in the functional requirements specification;
- there should be no any discrepancies between defined design requirements and developed design in the particular report;
- all attributes mapped properly, there are no any exceptions appeared during joining a valid attribute and measure. Each measure is calculated correctly with all possible attribute joined with. Each attribute joined with all other attributes and measures displays correct data.

4.2. Non-Functional Testing

Non-functional testing is the verification of the quality characteristics of the reporting solution. It refers to aspects of the software that may not be related to a specific function or user action such as security, performance, usability, compatibility or graphical user interface.

Test Case Techniques:

- access level verification - ensure that users from specified groups can access to specified reports:
    - only authorized users have access to the separate report area;
    - only users assigned to specified group have access to the report areas allowed for this group;
    - users cannot have access to the not allowed for this group report areas;
- web-browser security cases for the verification:
    - password should be encrypted during session;
    - browser back-forward buttons verification (does not break secure login);
    - unauthorized user should not be able to access pages he/she is not intended to;
    - sessions should time out after a specific time if user is not active etc.;
- join the most used set of attributes and measures related with the one separate report area, run report, measure report execution time and compare results with the time limitations defined at the performance requirements;
- join all possible attributes and measures (or the max data volume expected) related with the one separate report area, run report, measure report execution time and compare results with the time limitations defined at the performance requirements.

Quality and Acceptance Criteria:

- only defined in the Non-functional requirements specification users can have access to the PBI Dashboard reports;
- unauthorized user is not be able to access to the PBI Dashboard reports;
- only users assigned to specified group have access to the report areas allowed for this group;
- execution time of each report area is less than time limitations defined in the Non-functional requirements specification (TBD).

4.3. User Acceptance Testing

User Acceptance Testing is targeted on business users, who know their data and know how to use it. The purpose of UAT is to verify that reporting artifacts meet user's expectations and needs.

Test Case Techniques:

- check the correctness of the data compared to other available reports, if any;
- check graphical usability: layout, control elements (prompts, filers, sorts, drill-downs, links, etc.);
- check time of delivery, format of delivery, etc.

Quality and Acceptance Criteria:

- UAT of the PBI Dashboards will be conducted for each separate report area as soon as all testing activities are completed and responsible person accepts user story related to the delivered report area.

4.4. Regression testing

Regression testing should be performed to ensure that changes made to the PBI Dashboard do not cause any unintended issues or defects.

Test Case Techniques. Responsible team member (for example Integrity Manager) could be involved in listed bellow activities**:**

- verify each report area before deployment to prod;
- after new developments moving from the test environment into production;
- verify all modifications related with requested changes are deployed to prod (usually as a new features or change requests to make modifications in existing transformation logic);
- verify all modifications related with bug fixes are deployed (Confirmation testing) using test scripts;
- verify report areas after new developments moving into test environment (Smoke test). Test scripts will be used.

Quality and Acceptance Criteria:

- Results of comparison between PBI Dashboard report saved previously baselines (or results of SQL scripts execution written against EDW level which incorporates all transformation specified at the Mapping document) and PBI Dashboard report areas with applied changes should show no discrepancies between them.