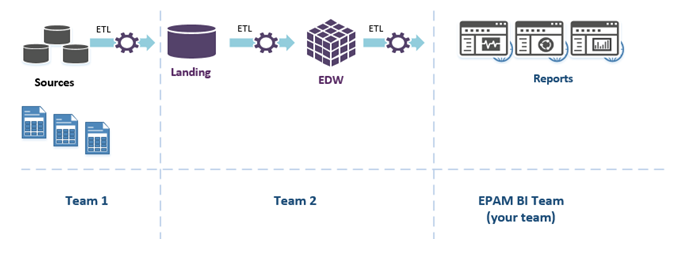


|  |
| --- |
| Data quality |
| DWBI Test Strategy Report |

# TEST STRATEGY



High-level project diagram

\*EDW – Enterprise Data Warehouse

## SCOPE OF WORK

The scope of work for the QA activities defined to the verification of the quality for 3 BI Power BI dashboards in the following priorities:

1. Power BI dashboard “Sales”
2. Power BI dashboard “Cost”
3. Power BI dashboard “Stocks”

The testing sequence for the Power BI dashboards will follow the priority set by the business needs and data dependencies. The **Sales** dashboard will be tested first, as it directly influences revenue tracking and decision-making. The **Cost** dashboard will follow, as it relies on **Sales** data for calculating expenses and profitability. Finally, the **Stocks** dashboard will be tested, as it presents supporting information on inventory levels but does not directly influence core business decisions.

## ENTRY CRITERIA

1. Necessary entry criteria required to start testing activities.

The QA team may suspend partial or full testing activities of the Power BI dashboards if any of the following conditions occur:

* **Data sources (DWH, ETL pipelines) are not properly configured or are unstable.**  
  Reports rely on structured and timely data delivery; without it, validation is not possible.
* **Power BI dashboards fail to open or visuals are not rendering correctly.**  
  This includes broken visuals, missing datasets, or dashboard crash during load.
* **Critical report functionality is non-operational.**  
  For example, slicers/filters don’t respond, drill-down fails, KPIs show errors, etc.  
  In such cases, a **"Smoke Test Failed"** status is applied.
* **Claimed functionality (new features, fixes, changes) is not included in the build provided for testing.**  
  No point in testing if the change request was not deployed.
* **New features appear, but do not work or work incorrectly.**  
  This makes regression or new feature testing invalid.
* **A critical issue blocks further testing activities.**  
  If the issue affects the ability to validate other areas, the testing is suspended until resolved.
* **Previously reported blocking issues are not resolved.**  
  QA cannot proceed without addressing previously acknowledged blockers.
* **A new build/version of the dashboard is made available for testing.**  
  Current testing may be suspended to switch to a newer, more relevant version.

**Note:** The decision to continue testing despite the conditions above will be made by the QA Team Lead or designated QA Engineer based on the test scope, current priorities, and availability of stable components.

1. Possible risks.

| **#** | **Risk** | **Severity** | **Risk Description** | **Resolution** |
| --- | --- | --- | --- | --- |
| 1 | Data sources are unstable or not ready | High | DWH or ETL pipelines contain errors or data arrives with delays. BI dashboards display incorrect or outdated data. | Postpone test start until sources are stabilized. Monitor ETL. Clarify with the DWH team. |
| 2 | Dashboards do not contain the expected changes | Medium | The Power BI reports do not reflect the latest features or fixes. | Verify build version. Regular communication with the development team. Perform smoke testing before deeper QA. |
| 3 | New features are unstable or only partially functional | High | It is not possible to perform functional testing on new dashboard components. | Escalate to development. Pause testing of the feature. Use workarounds. Prioritize bug fixes. |
| 4 | Frequent requirement changes near the testing deadline | High | Changes to business logic or visuals are introduced too late, making test cases outdated. | Set a freeze period 3 days before release. Track requirements in Jira. |
| 5 | Lack of feature documentation | Medium | Missing detailed description for new logic — unclear expected behavior of new visualizations. | Request technical documentation. Consult with analysts. Perform manual verification if needed. |
| 6 | BI team is dependent on other teams (ETL, DWH) | High | Errors in upstream processes block testing activities on BI side. | Improve communication between teams. Assign dedicated contacts. Establish cross-team sync. |
| 7 | Business metric logic errors | High | Sales, costs, or stock data may appear incorrect — the issue may lie in calculations, not visuals. | Cross-check with reference data. Perform manual spot-checks. Involve business analysts if needed. |
| 8 | Scope is increased by new unplanned Features or Change Requests | High | Iterations Scope is increased.  Delays in implementation date. | Efficient negotiation and approval of the impact to schedule/quality caused by the delay signed off by Customer.  Possibly reduce increased scope by dropping low-priority Features. |

## TEST APPROACH

The testing approach for the Power BI dashboards (“Sales”, “Cost”, and “Stocks”) will be executed in the following stages:

1. **Review of Functional Specifications**
   * Analyze business and technical requirements for each dashboard
   * Clarify unclear or missing acceptance criteria with stakeholders
2. **Test Planning**
   * Define the test scope, required resources, timelines, and responsibilities
   * Prepare the test strategy and entry criteria
3. **Test Case Design**
   * Design high-level test scenarios based on the functional specs and data models
   * Define validation rules for KPIs, filters, and business logic
   * Identify critical data quality checks and edge cases
4. **Test Environment Setup**
   * Ensure dashboards are deployed in QA workspace
   * Validate access rights, data refresh schedules, and connectivity to DWH sources
5. **Functional Testing**
   * Validate layout, filter logic, and visualization behavior
   * Verify KPIs and calculations match expected business logic
   * Confirm drill-down and cross-filtering behavior works as intended
6. **Data Quality & Backend Validation**
   * Compare dashboard data with source systems (SQL, DWH)
   * Validate data aggregation, transformations, and currency logic
7. **Non-functional Testing**
   * Performance checks (e.g., load time of visuals)
   * Security/access testing (role-based access verification)
8. **Defect Logging and Retesting**
   * Report found issues to backlog (e.g., in Jira)
   * Retest fixed issues and perform regression where needed
9. **Test Closure**
   * Summarize testing progress and results
   * Prepare the Test Result Report with visualized metrics