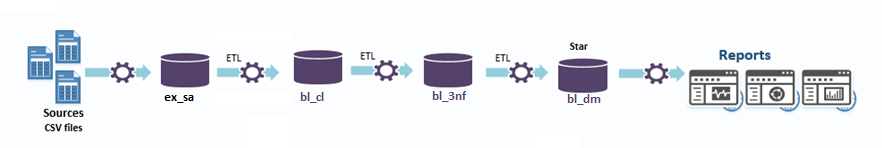


|  |
| --- |
| EPAM Systems, RD Dep. |
| Exit Task |

# TEST STRATEGY



High-level project diagram

## ENTRY CRITERIA

Entry criteria:

* Developed solution deployed to the test environment
* Test Environment and database are available
* Test Schedule is completed
* Source data is available

Risks:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk** | **Severity** | **Impact** | **Possible Resolution** |
| 1 | Source data is not provided in time | High | Delays in implementation date;  Loss of the suitability of the test documentation | Create a synthetic dataset which will help to start testing procedures without the actual data |
| 2 | Test environment malfunction | Critical | Delays in implementation date;  Possible data loss. | Point out the source of the malfunction (hardware, software, human factor) and dispatch appropriate team to mitigate the impact.  Back the data up on another on-premise solution |
| 3 | Development team members sick leace | Critical | Since dev team consists of 1 man, it puts all development, testing and maintenance on hold for indefinite amount of time. | Hire additional team members |
| 4 | Inconsistent data | High | Some records could be lost and/or loaded wrongly into the solution due to high fragmentation of the raw data | Reword original data source data extraction system if possible;  Implement additional cleansing procedures |

## TEST APPROACH

Since our solutions data have already been stripped of all sensitive and/or personal information we wouldn’t need to develop and test any security system, because data leakage would not be an issue since all data is already in public access and could be used for analytical purposes only. Implementation of such system would only further raise the price of development and maintenance and require additional time for implementing.

Since our solution consists of multiple layers and developed by a trainee we would perform a thorough Smoke Testing which will include manual comparison and comparison via SQL scripts of tables’ metadata (attribute names, constraints, datatypes and etc.), fact table initial load and incremental load testing, SCD2 procedures testing.

After Smoke Testing we will commence a Data Validation testing, which will include:

* Data Completeness Verification - data comparison between all contacting layers of the solution (ex\_sa - bl\_cl, bl\_cl – bl\_3nf, bl\_3nf – bl\_dm and so on)
* Data Transformation Verification – manual data formatting and transformation comparison and SQL query comparison, calculation results verification and so on
* Data Quality Verification

After Data Validation is complete it would be best to proceed with reports’ GUI testing to ensure business valuability of those reports and completeness of those reports data, check if reports give user basic information about business current status.

## HIGH-LEVEL CHECKLISt

|  |  |
| --- | --- |
| **DW/BI Layer** | **Validation Description** |
| Back-end | Check if all schemas(users) exist  Check if all tables exist  Check if all the necessary right are granted  Check if each users rights are limited ONLY to the rights provided in the documentation  Deploy synthetic test data from .csv file all the way to reporting area  Duplicate record loading from the ex\_sa all the way to bl\_3nf  Attempt loading data to a layer which is not in direct contact with the target layer  Attempt loading data with the same PK and different data to non-SCD tables via procedures  Incremental load testing  Initial load testing  Wrong dataset name input in 3NF layer procedures  Wrong dataset name input in DM layer procedures  Rollback after  Check reports RLS  Check reports calculations accuracy |
| Front-end | Check reports’ visualizations displaying according to design requirements  Check reports’ resolution  Check appropriate color schema  Check if reports comply with guidelines |