**Title: Test approach for GDD MDM**

**GDD MDM**

**Project Clarity ID: 080596**

**Test Phases and Responsibilities**

**Testing includes the below scopes:**

1. EBX5-User Interface Testing
2. Data Migration Testing
3. System Integration Testing

The table below gives an overview of the applicable test phases including on which system environment those tests will be performed as well as the responsibilities for the different test phases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Phase** | **Environment** | | | **Responsible** |
| **DEV** | **Test / QA** | **PROD** |
| Functional/Integration Testing (OQ) | X (Informal) | X (Formal) |  | Testing Factory team |
| Data Migration testing | X (Informal) | X (Formal) |  | Testing Factory team |
| User Acceptance Testing (UAT) |  | X |  | BA / Business Lead / MDPM |

# Test Strategy

**EBX5-User Interface Testing**

The verification of the system functionality in EBX5 UI as outlined in “GDD\_R1\_Functional\_Specification- v1.3” will be covered as part of this scope

1. **Data model validation** – Each table and its fields will be displayed in the UI. Each table fields and properties displayed will be validated against the properties defined in “GDD - R1 - Detailed Business Glossary” with admin access. or properties file will be shared by Dev team and this will be validated against “GDD - R1 - Detailed Business Glossary”
2. **Work flow validations** – Approach is to Create, update, deactivate and reactivate existing / newly created data through EBX5 and validate all the work flows as mentioned in the functional specifications. Validation of audit trial will be done as per the functionality mentioned in FS for all CRUD operations
3. **Import and Export validations** – Approach is to validate all kinds of formats for import by creating different sets of data for each import. Export feature is validated against mentioned format in FS.

**Assumptions:**

* + - DB access will not be provided to testing team
    - Dev team will share extract of properties file to validate data model
    - Since UI, validation is prior to data migration; dummy data will be loaded in Dev and QA environments by Dev team

**Test Data:** New test data will be created through UI for validations and same test data will be updated and deactivated for few test cases. Existing data will be used for modifications for few other validations

*Environment: Dev for informal testing and QA for formal testing*

**Data Migration:**

**Pre-Migration Testing:**

1. Validate all the tables/columns have been implemented as per data model.
2. Validate the columns are implemented with correct data type, length as per data model
3. A number of records in each table should be noted in the legacy system.
4. Study the interfaces in the new application and their connections. Data flowing in the interface should be highly secured and not broken.
5. Prepare test cases, test scenarios, and use cases for new conditions in the new applications.
6. Execute a set of test cases, scenarios with a set of users and keep the results, logs stored. The same needs to be verified after Migration to ensure that legacy data and functionality are intact.
7. Count of the data and records should be noted down clearly, it needs to be verified after Migration for no loss of data.

**Migration Testing :(I think this will be out of scope)**

1. Validate all the jobs are running successfully.
2. Audit information should be captured for the transformation of the data.

Ex: In the legacy system, the currency is mentioned as per the source country and in new system if the currency needs to be updated in unique code.

1. Error Log should be captured for the loss of data (Primary key violation, not null check etc.….)
2. Connectivity between all the components of application is checked

**Post Migration Testing (Business Rule Validation)**

1. Check whether the source file is segregated into multiple target files as per the requirements.
2. Check whether all the data in the legacy is migrated to the new application within the downtime that was planned. To ensure this, compare the number of records between legacy and the new application for each table and views in the database.
3. Data migrated from the legacy to new application should retain its value and format unless it is not specified to do so. To ensure this, compare data values between legacy and new application’s database.
4. Check whether the data is transformed and updated correctly in the new system.
5. Any data addition in the new application should not reflect back on the legacy
6. Deleting the legacy application’s data in the new application should be supported. Once deleted in the new application, it should not delete data in legacy as well.
7. Carry out functionality related tests with a variety of data samples (different age group, users from different region etc.,)
8. It is also required to verify if ‘Feature Flags’ are enabled for the new features and switching it on/off enables the features to turn on and off.

**Out of scope:**

1. Performance testing is important to ensure that migration to new system/software has not degraded the performance of the system.
2. It is also required to carry out Load and stress tests to ensure the system stability.
3. Verify that the software upgrade has not opened up any security vulnerabilities and hence carry out security testing, especially in the area where changes have been made to the system during migration.
4. Usability is another aspect, which is to be verified, wherein if GUI layout/front U end system has changed, or any functionality has changed, what is the Ease of Use that the end user is feeling as compared to the legacy system.

*Environment: Dev for informal testing and QA for formal testing*

**Integration Testing:**

The verification of the system integration functionality as outlined in “IMPACT\_MDM\_INT Functional Specification-Inbound\_v0.2” will be covered as part of this scope

1. **Investigator Creation and Update thru external source:**

- Creation/Update request will be triggered using SOAP UI with appropriate parameters.

- Workflow will be verified for the desired record in EBX5

- Landing table will be verified in EBX5 to make sure the table has anticipated record

* MDM will push the record to Boomi.
* Boomi team will provide the JBoss MQ and Boomi log files that has the data written to it
* Log files will be validated against the records, which has created and pushed from MDM.
* Creation/Update request to Remote Keys table will be triggered using SOAP UI with appropriate parameters
* “Remote keys” table will be verified to make sure the table has desired “Global and Consumer IDs”

1. **Delta record validation:**

* CRUD operations for Site and Investigator will be performed in EBX5
* Workflow will be verified for the desired records in EBX5
* MDM will push the record to Boomi.
* Boomi team will provide the JBoss MQ and Boomi log files that has the data written to it
* Log files will be validated against the data, which has created and pushed from MDM.
* Creation/Update request to Remote Keys table will be triggered using SOAP UI with appropriate parameters
* “Remote keys” table will be verified to make sure the table has desired “Global and Consumer IDs”

**Assumptions:**

* N/A

**Test Data:** New test data will be created through MDM UI and external source for validations and same test data will be updated and deactivated for few test cases. Existing data will be used for modifications for few other validations

*Environment: Dev for informal testing and QA for formal testing*

The OQ & UAT test specifications shall be created, reviewed and approved before execution in PROTON test management tool and the execution shall be performed/approved & results shall be stored within PROTON.

For this project, all test specifications will be documented in the Proton tool.

User requirements as well as functional requirements will also be loaded in the tool.

Tests will be executed and the results documented in the Proton Tool itself.

All defect/deviation will also be logged and tracked in the tool itself.

After completing the execution of the testing, all evidences will reside in the Proton tool, from test specification, to test execution, to defect lists and even traceability matrix.

All team members involved in the testing, in Proton shall be trained on the tool.

Informal testing will be done in the development environment.

All teams need to agree on the usage of the test data in that any testing will naturally alter the test data files therefore coordination is essential.

All documented testing will be done in the QA environment.