**Modern Data Platform**

**Functional Test Strategy/Approach**

Version: - 0.3

Document Control

| **Version** | **Dated** | **Author** | **Reason for revision** |
| --- | --- | --- | --- |
| 0.1 | 21/09/2022 | Ram Baskar | Draft/Issued for review |
| 0.1 | 28/09/2022 | Ram Baskar | Added pictures in multiple sections |
| 0.2 | 10/10/2022 | Ram Baskar | Incorporated comments after initial walkthrough about E2E & Test Data |
| 0.2 | 12/10/2022 | Ram Baskar | Appendix included with defect management life cycle & check points |
| 0.3 | 20/11/2022 | Ram Baskar | Updated context diagram |

# Sign-off *(to be updated later)*

| **Name** | **Role** | **Sign Off Date** |
| --- | --- | --- |
| Adam Nowacki | Head of Data Engineering |  |
| Ricardo Pedro | Enterprise Data Architect |  |
| Anil Bangaru | Data Architect |  |
| Graham Binner | Data & Reporting Architect |  |
| Abhishek Gupta | Lead Data Engineer |  |
| Akhila Mathews | Senior Business Analyst |  |
| Will Tomlinson | Cloud Operations |  |
| ??? | IT Project Manager |  |
| ??? | Business project Manager |  |

# Distribution for review *(to be updated later with more team members from MDP)*

| **Name** | **Role** |
| --- | --- |
| Kanika Chaganty | Chief Data Officer |
| Maheswari Ramasamy | Data Manager |
| Ram Baskar | Lead QA Engineer |
| Adam Suchley | Technical Delivery Lead |
| Sumit Thakur | Test Analyst |
| Anna Drozynska | Lead Data Engineer/Scrum Master |
| Michal Nowocien | SME – Data Engineer |

Contents

[Sign-off *(to be updated later)* 3](#_Toc120608611)

[Distribution for review *(to be updated later with more team members from MDP)* 4](#_Toc120608612)

[1 Introduction 6](#_Toc120608613)

[1.1 Background 6](#_Toc120608614)

[1.2 Purpose 6](#_Toc120608615)

[1.3 Document Objectives 6](#_Toc120608616)

[2 Scope 7](#_Toc120608617)

[2.1 In-Scope 7](#_Toc120608618)

[2.2 Out of Scope 7](#_Toc120608619)

[3 Test Governance 8](#_Toc120608620)

[3.1 Software Test Life Cycle (STLC) 8](#_Toc120608621)

[3.2 Agile Methodology 8](#_Toc120608622)

[3.3 Test Approach & Techniques 9](#_Toc120608623)

[3.4 High Level context diagram – Data Ingestion/process in MDP platform 9](#_Toc120608624)

[3.5 Test Phases 10](#_Toc120608625)

[3.6 Quality Gates & Entry/Exit Criteria 11](#_Toc120608626)

[3.7 Defect Life Cycle 11](#_Toc120608627)

[4 Resource 13](#_Toc120608628)

[4.1 Test Data 13](#_Toc120608629)

[4.2 Tools 13](#_Toc120608630)

[4.3 Automation 13](#_Toc120608631)

[4.4 Roles and Responsibilities 14](#_Toc120608632)

[4.5 Test Environments 15](#_Toc120608633)

[4.6 Skills and Resource Requirements 15](#_Toc120608634)

[4.7 RIDA (Risks, Issues, Dependency & Assumptions) 15](#_Toc120608635)

[4.8 Reporting & Statistics 16](#_Toc120608636)

[4.8.1 Test Progress Reporting 16](#_Toc120608637)

[4.8.2 Test Completion Reporting 16](#_Toc120608638)

[5 Glossary 17](#_Toc120608639)

[6 Appendix 18](#_Toc120608640)

[6.1 Appendix A – Defect/Bug Life Cycle 18](#_Toc120608641)

[6.2 Appendix B – Sample Check Point 19](#_Toc120608642)

# Introduction

## Background

The Modern Data Platform (MDP) is an Azure hosted data repository managed by Cloud Data Platform team. The new architecture will replace the existing platform called “MYMI” (also running in Microsoft Azure Platform) with design improvement and aiming to reduce the operational cost. It aims to integrate the data from different sources available in “on-premises” databases and provides consistent information to different user groups across the business, for use in analytics, reporting, data extract and data processing activities.

It is intended to provide all the project team members with the overview of functional testing processes that will be adopted for BRIT Modern Data Platform. This document does not intend to comprehensively detail what the coverage of testing will be, for each project/use case, a separate test plan will be created with this level of information.

## Purpose

The purpose of this document is to define how the common quality/acceptance criteria is accomplished to ensure that, the final delivery of the processed/transformed data in MDP environment is functionally fit for purpose for consumers/business user group to access the data via portal, visualisation tools like (SSRS/Tableau/Power BI) and data extract etc.

## Document Objectives

* Define test objectives, In-scope/out of scope functions and corresponding test items to clearly understand what areas need to be tested
* Outline how the test approach will be followed to ensure all parties understand the testing stages and corresponding timelines including test automation where possible
* Define an overall testing process/activity that ensures how the whole Modern Data Platform is tested to the degree required for consumers/business to accept the delivered system into services
* Define entry/exit criteria
* Detail the type/phase of testing e.g., SIT, UAT, NFT etc. that will be conducted to ensure clarity on how relevant test phases will be tested
* Lists the environment(s) and test tools in consideration
* Document the clarity on the roles and responsibilities throughout the testing activities
* Define defect management life cycle

# Scope

## In-Scope

|  |  |
| --- | --- |
| **Functional Area** | **Description** |
| Data Extraction & Ingestion  (Via Batch Ingestion) | Extraction, Ingestion & Transformation of data available in on-premises database/file (*CSV/TEXT/XML/JSON* etc) and/or from other cloud like *AWS/GCP* etc *(not applicable at the moment)* into MDP Platform |
| Data Extraction & Ingestion  (Via Streaming) | Extraction, Ingestion & Transformation of data coming from streaming like Event Hub/Kafka etc. into MDP Platform. *N/A at the time of creating this document* (future design) |
| Landing Zone  (ADLS Gen2) | Verification of data landing in Parquet/Delta format in an appropriate container/folder across all the available layers (Bronze/Silver/Gold) |
| Data validation  (Azure Synapse) | Data validation between source – target across multiple layers within Data Platform including reconciliation, data completeness, data correctness and data transformation based on transformation rules defined in *data mapping*  *(This includes Reference Data (if any)/Transaction/Dimension/Fact and Power BI report contents including measures)* |
| Outgoing Data Extract | Validate the data extract from MDP to 3rd party based on the business requirement/rules defined. *Example Riverstone extract* |
| Other Functions (based on current MYMI process) | Data validation/reconciliation for Month/Quarter-end partitions  Financial adjustment (if applicable) including Permanent/Temporary |
| Non-Functional Testing | Performance of all components within MDP Platform like   * Volume & Performance * Azure Synapse Data Pipeline * Data extract to 3rd party * Function App Event Hub/ Kafka(future) – *N/A at the moment* |
|  |  |

## Out of Scope

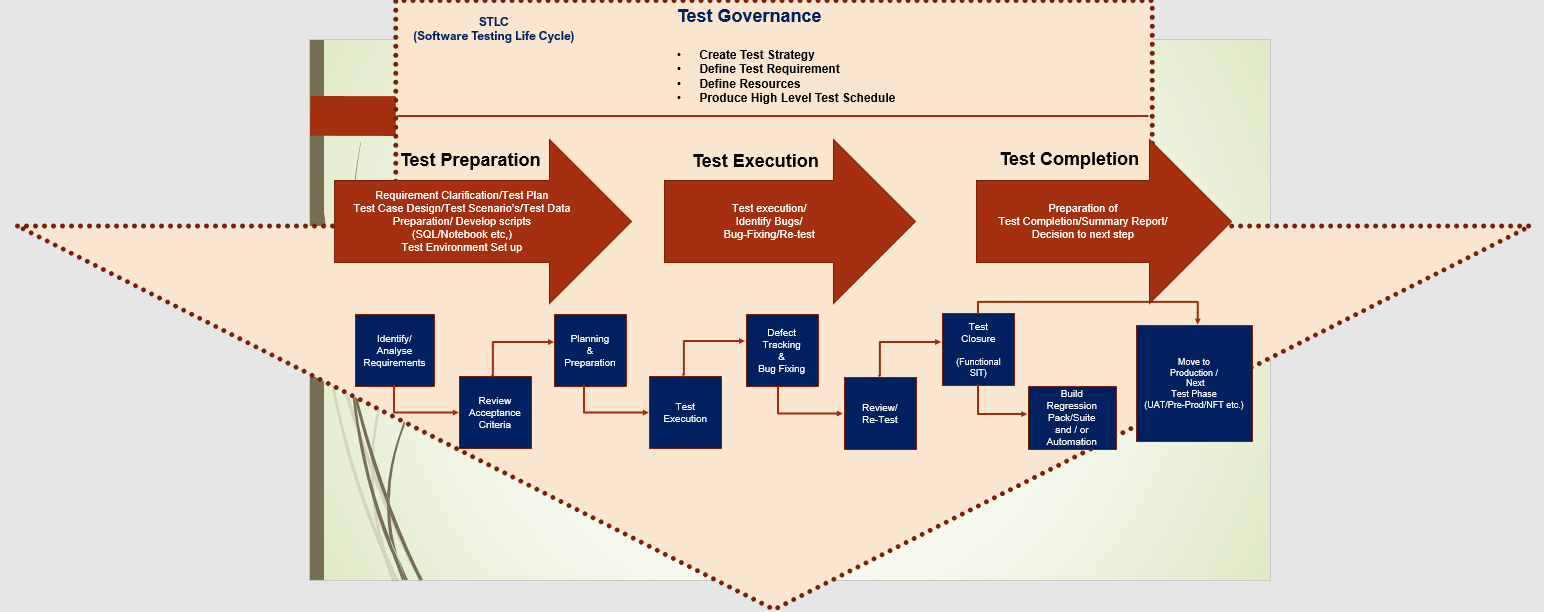
|  |  |
| --- | --- |
| **Area** | **Description/Justification** |
| Data Quality | The source system owner is responsible for source data quality |
| Security - Infrastructure | BRIT security team is responsible for infrastructure related security |
| Front-end applications/API | There is no requirement for front-end/web/API related testing in MDP |
| Latency | Latency outside MDP platform will be tested either by dedicated NFT or by Cloud operations team |
| Downstream Applications | Applications outside MDP Platform like portal etc. Analytics related like ML/AI Module |
| Non-Functional | Infrastructure / Security / Penetration /Alert / Monitoring / Fail-over / High Availability /Backup-restore etc. (Possibly scope for either NFT team/Cloud Operation team) |

# Test Governance

## Software Test Life Cycle (STLC)

**STLC** is a set of specific steps/activities used during the testing process to ensure software quality gates are met to satisfy the acceptance criteria/requirement. Testing is a critical part of preparing software for use, and a STLC helps make this process more sophisticated, consistent and effective. It involves lot of preparation, verification and validation activities and is not just a single isolate activity. Many STLC setups start with an analysis of testing requirements, followed by test preparation which includes test estimation, planning and test data preparation. Later, there is test execution and other follow-ups, such as regression testing for more complex functionality and retesting some of the elements based on the bug/defect found during test execution.

The fundamental approach and structure of testing undertaken by MDP QA team is detailed below. Planning & Control is used for overall of each use case/project and the remaining activities are applied to each individual test phase.



## Agile Methodology

The MDP QA team will use the elements of the Agile project management methodology for all stages of E2E SIT testing. The UAT and Pre-Prod approach (if applicable) will subject to further discussion and may undergo further changes in ways of working. QA team will actively contribute to all the agile related events like planning, refinement, retro and sprint review. Scrum is the most popular framework that brings effective collaboration between different teams working on complex project/product. We follow **Agile Scrum Methodology**, which is sprint-based system for managing software projects that works on incremental development in project development. The iterations are divided into manageable time slabs, each of about two-week sprints. Every goal-driven sprint offers vital features that constructively develop a complete product/feature. In subsequent sprints, add-on features, and enhancements based on customer and team feedback are built into the product releasing the fully tested feature/software/product every 2 weeks *(to be detailed as part of Release Strategy).*

## Test Approach & Techniques

Testing Modern Data Platform is different from traditional application testing as it requires a data driven approach. The standard approach is to consider ETL/data pipeline/data ingestion process using Microsoft azure tools like ADF/Azure Synapse etc. as black box. The data extracted from the on-premises source database tables/files etc. will be moving through various stages across the multiple layers ***(bronze/silver/gold)*** in cloud platform based on data mapping document. The data validation process is to verify the data at each stage including landing/transformation/aggregation for different type of entities like Dimension & Fact etc. and to ensure that, transformed data is fit for the purpose for consumers to access the data via data extract/portal/mobile devices (if any)/visualisation tools like Tableau/Power BI etc for reporting and analysing purposes.

Below is the list of high-level activities for functional E2E system integration testing phase

* *Walkthrough of feature/requirements/functionality between Dev/BA to QA team via 3-amigos call*
* *Based on the agreed acceptance criteria and data mapping document, QA team will perform the test preparation including estimation, design test case, develop script and arrange test data while the feature is being developed*
* *Review test cases internally to ensure that test coverage is as complete as possible and get tests ready for execution*
* *Once the code is deployed in Test Environment via CI/CD pipeline, facilitate the team to work collaboratively to perform test execution (System Test / E2E SIT) and identify the defects in the early stages*
* *Aim to have SIT testing completed and sign-off completion report prior to promote the code to next test phase/PROD*
* *Build a regression pack for future test execution and automate tests where possible*

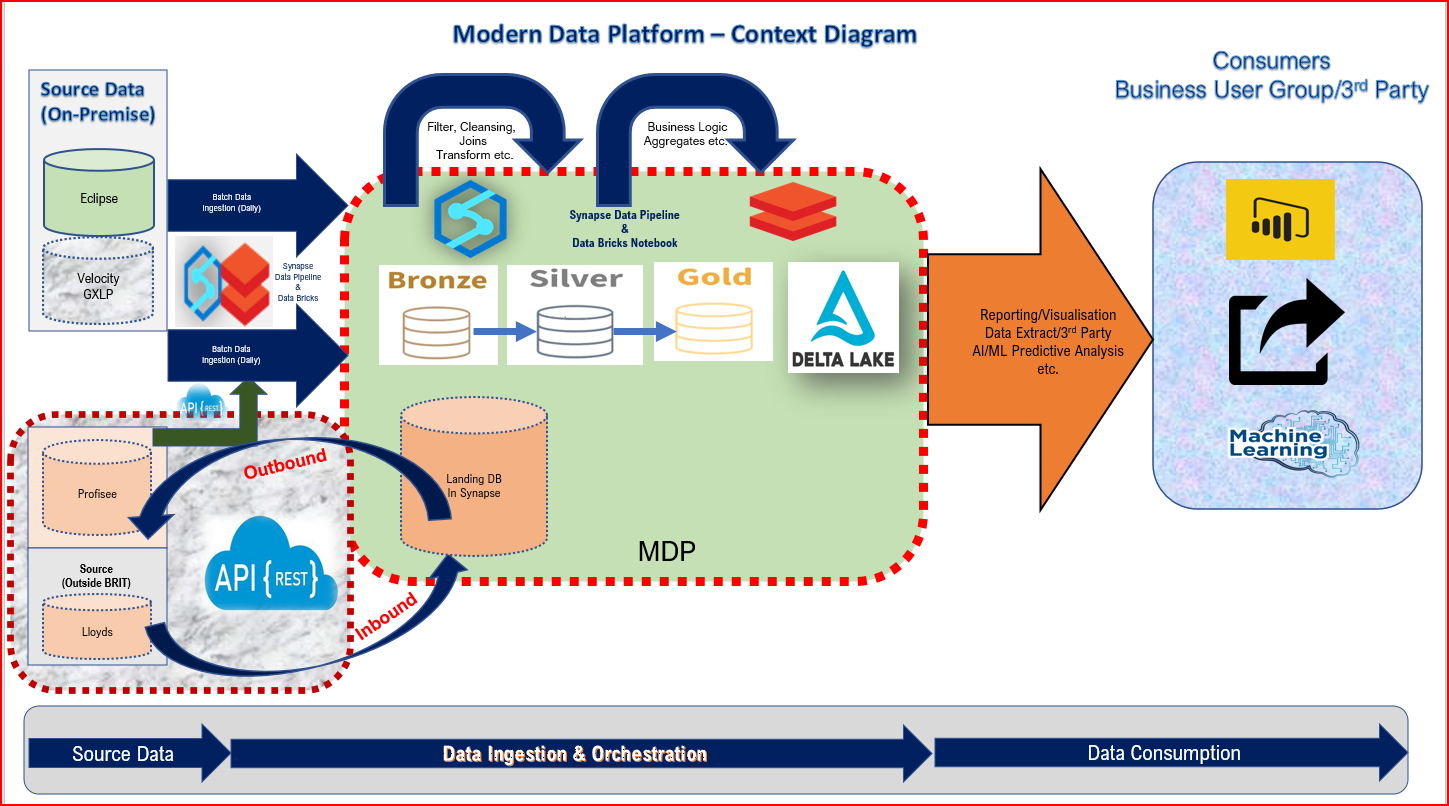
To facilitate the coverage of requirements, functional tests will be executed with the multiple data sets from source to make sure that, all the relevant scenarios are covered

Test execution during E2E SIT phase will be covered with at least 2 different data sets.

* Initial Load followed by delta/incremental for Reference data (if applicable) & Transactional data
* Month-end / Quarter-end process data validation including boundary value testing based on partition *(if applicable)*
* Validate Power BI report based on the reporting requirement/model and track the details/measures all the way back to source data
* Data extract from Final destination (gold layer) to 3rd party (example for Riverstone)

For functional testing, multiple check points will be defined based on LDM/PDM (if available) and verified at each step where the data is moved from source (on-premises) to cloud environment across multiple layers. Refer *Appendix B* for a sample checkpoint

## High Level context diagram – Data Ingestion/process in MDP platform



## Test Phases

The test phases identified for Modern Data Platform are listed below. Each test phase should be organised and executed using the STLC approach. The test phases are executed in the order listed but do not necessarily need to be run in synchronisation between individual software/component deliverables until the integration and E2E test phases are reached.

The test phases have been created with the objectives of.

* Confirming each level of testing/delivery is successful
* Supporting the following phase
* Collectively supporting the E2E test phase and acceptance of the final product/feature
* Achieving delivery within sprint goal

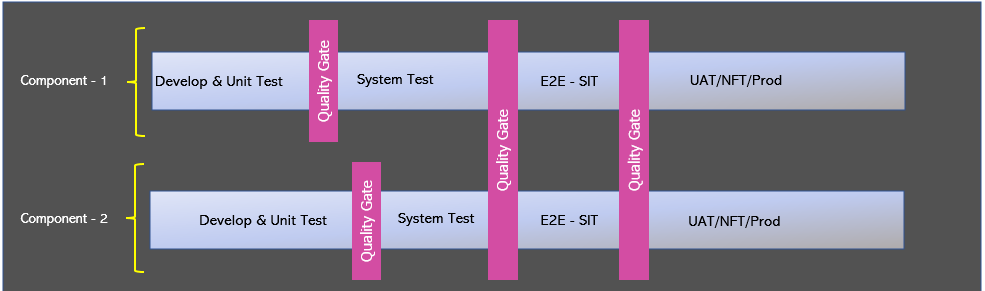
|  |  |
| --- | --- |
| **No.** | **Test Phases** |
| 1 | Static Testing |
| 2 | Unit Testing |
| 3 | System Testing (a specific source/layer) |
| 4 | E2E SIT System Integration Testing (multiple source/layers) |
| 5 | UAT (acceptance test from business) |
| 6 | NFT (Non-Functional Testing) |
| 7 | RFO/OAT/Production/Commission Testing) |

All the above test phases are mandatory and is highly recommended from long term strategy perspective. However, for each feature MDP QA team will identify early if there is any need for these and liaise with the relevant stakeholders and obtain an agreement accordingly. In short term, ***Dev-Test-Prod*** approach will be in place until we built more features in MDP

Test Phases should be run following a linear model. Each test phase should only commence once the previous test phase quality gate has been achieved based on the measurable metrics. By following this approach, it ensures: -

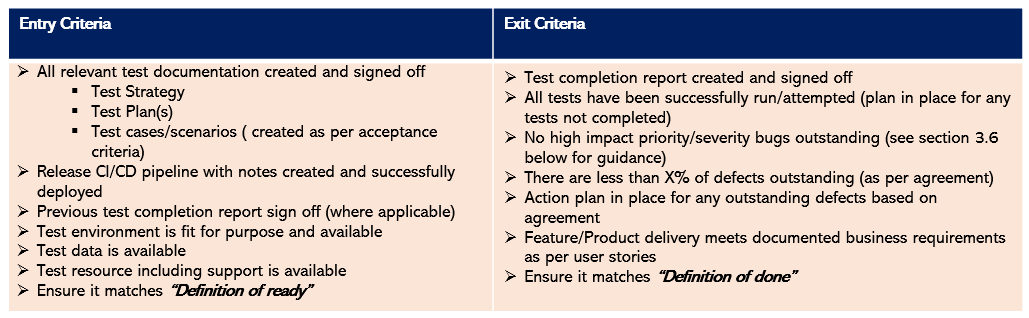
* The overall plan is being adhered to thus ensuring timescales are met
* The test phase is ready to commence
* System/Product/Feature quality can be tracked through STLC
* Defects are found in early stage make it easier, quicker, and cheaper to fix
* Test coverage is achieved in the minimum/agreed timescale

Test phases for individual components do not need to be in sync until SIT phase is reached. At this point all components/sources that make up the feature/product need to be ready for Integration testing



## Quality Gates & Entry/Exit Criteria

Quality Gates are generally positioned between each test phase. There should be a separate set of criteria to measure completeness of the delivering test phase (exit criteria) followed by a measure of the readiness to move into the subsequent test phase (entry criteria). The table below provides a generic list of what will be contained within the quality gates



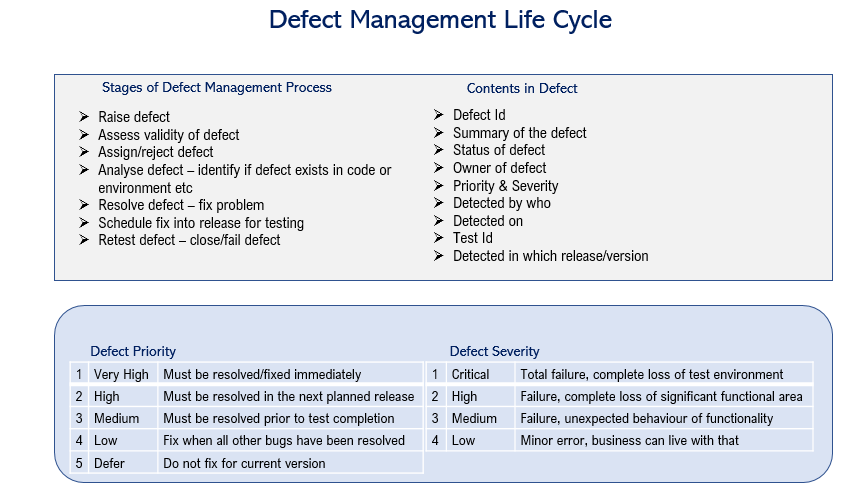
Clear, measurable quality gates must be established and agreed before testing commences to ensure both progress and quality can be quantified throughout the project. At the start and end of each test phase the criteria should be assessed to ensure it has been met before moving into the next test phase or accepting delivery of the final feature/product. If the criteria have not been achieved, then the next test phase should not commence until all outstanding criteria have been addressed.  If quality gates are not implemented or adhered to then this will result in several outcomes as below: -

* Slow test progress
* Late identification of bugs (which take longer to analyse/fix/retest and expensive)
* Final quality of the delivered feature may not meet business requirements/acceptance criteria

Each test plan should contain explicit entry/exit criteria relevant to the particular test phase.

## Defect Life Cycle

Once the SIT test phases have been reached daily defect meetings will be held to review all outstanding and new faults. The meeting will also be used to provide a general update on the progress of testing and to schedule any fixed defects to be released into test. It is expected that all the relevant stakeholders are expected to be at the meeting. Basic information that will be provided in each defect as part of defect management life cycle is listed below. Refer **Appendix A** for defect management life cycle flow chart



# Resource

## Test Data

One of the main deliverables to start E2E SIT is the test data and will be sourced from the relevant on-premises source system(s) based on the meta data information populated in the control framework database. The pre-requisite to start E2E SIT is to make sure that, all the individual component quality gates are met from system testing phase (if multiple source systems were involved for the use case/business requirements). MDP QA team will work with the business stakeholders/business analyst(s) during test preparation to identify the test data required for the items being tested to ensure that sufficient data is available to satisfy all the scenario’s/test conditions. MDP QA team will request the relevant source system owner(s) for test data either cut down version or full copy (based on the individual use case/test phase requirement) for an initial load, followed by an updated data for an incremental/delta load.

In some instances, test data can be created by the MDP QA team without comprising the testing, but the preferred option is to always use real test data/production-like data. If it is not possible to create / source test data due to time constraints on any test phase, then a risk assessment takes place.  A caveat is put in place and real test data must be used in the next following test phases/environments. For example, if there is test data on SIT, but not UAT or PRE, then it is acceptable to sign off with a caveat as at this stage, we are just testing the process works.

MDP test team is also responsible for extracting and processing the data across the multiple layers and eventually transformed and loaded into gold layer within Data Lake in both Parquet/Delta format. MDP QA team is also responsible for validating the data including reconciliation (where possible) between source to target in all layers starting from *bronze – silver - gold*, data correctness, data completeness, data transformation and data integrity across the entities. Low-level test details will be provided in a separate document (Test plan document) per use case/project including all the scenario(s)

*It’s highly recommended to use the production-like data for any use case/project in the event of any requirement or to prove the functionality is not broken as per the current MYMI architecture in MDP platform by doing like-for-like comparison where possible like Power BI report etc.*

## Tools

MDP QA team will continue to use ADO for test management, test execution and defect management. All the test cases for SIT and Regression Pack will be designed/developed in ADO. Each test case will be linked to user story within the project/use case from coverage perspective. For the earlier test phases whoever is leading should use whatever existing processes are in place.

It is recommended that all defects identified across all the test phases are managed using ADO. This test tool can only be accessed internally to BRIT team

The defect management process is further explained in section 3.6 of this document and appendix A

## Automation

The automation process within MDP will be a work-in-progress activity and the key area to automate is for data validation between source – target entities (based on data mapping/transformation rules) across all the layers. To start with manual testing during E2E SIT phase, regression pack scripts will be automated (where possible) to achieve the benefit of running the tests quickly with the aiming to achieve 100% automation. However, tests scripts will be automated during E2E SIT phase where possible to run the tests quickly and identify the bugs early

High Level approach of automation: -

* Stand-alone T-SQL/CTE scripts will be designed to validate data including transformation/reconciliation and test results will be captured in QA database with pass/fail status for individual record per entity being tested
* Once the script works as expected for manual testing, convert stand-alone scripts into stored procedure
* Develop Azure synapse pipeline to call the stored procedure related to QA
* With the help of Cloud Ops team, create CI/CD QA pipeline to deploy/run automated tests in the subsequent environment/test phases

## Roles and Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| ???? | * Sign-off Test Strategy * Sign-off Test Plan * Sign-off Test Schedule * Sign-off Test Completion Report(s) |
| LTE/TM  (Lead Test Engineer /  Test Manager) | * Lead the test team and manage testing tasks * Write the test approach and ensure it is followed at all stages * Undertake relevant test phase sign-off * Liaise with specialist teams (Performance, Security etc.) to incorporate non-functional testing into the test approach * Plan the daily testing workload and reporting of progress * Remove any blockers that occur during testing |
| TE  (Test Engineer) | * Define test coverage and prioritise tests * Design test cases and execute tests * Ensure all defects are allocated, fixed and retested * Actively update regression pack for future regression |
| Performance TE  (PTE) | * Write the performance test plan in conjunction with the LTE/TM * Execute performance testing and assist with test sign off * Work with the project team to ensure that security and OAT testing (where required) is executed and signed off |
| Business/Data Analyst  BA/DA | * Contribute to and review test artefacts * Joint responsibility with the LDE/SDE for reviewing and signing off test coverage |
| Data Engineer  DE | * Execute unit testing and provide evidence * Joint responsibility with the BA/DA for reviewing and signing off test coverage |

## Test Environments

Cloud ops team will be responsible to define and manage test environments. There are a number of assumptions associated with the provision and support of test environments which are detailed below: -

* Test environments will be provided for the purpose of E2E SIT followed by other test phases
* Full uninterrupted access will be available to all MDP QA team members (apart from the need for new releases or agreed outages)
* It is expected that all test environments will be supported in line with existing production support procedures Only Production issues should supersede the support for test environments
* Any issues with test environments will be logged and managed using the defect management process as detailed within this document (as per Appendix A)
* Controlled Release Management process will be followed as per *Release Strategy*

Summary of environments as below (To be discussed)

|  |  |
| --- | --- |
| **Environment** | **Description** |
| DEV | Development + Unit Test |
| TEST | E2E SIT Functional + Component Integration (within MDP) + NFT *(In-scope)* |
| UAT | Integration Testing (MDP + Portal/Visualisation Power BI/Tableau etc.) |
| Pre-Prod | Non-Functional Test based on NFT requirements + Functional Process verification |
| Prod | RFO – Ready for Operation (Data Reconciliation) |

## Skills and Resource Requirements

The MDP QA team would need to have the below detailed skill sets:

* ETL / Business Intelligence / Enterprise Data Warehouse Testing experience
* Knowledge of designing Test Case, System testing, Regression testing
* Knowledge of Test management tools like ADO/JIRA etc.
* Good knowledge of tools like SSIS/SSRS/Power BI, Tableau
* Good understanding about Azure Cloud Technology (ADF/Synapse/Data Lake/Delta Lake/Data Bricks etc.)
* Good understanding of data streaming technology like Kafka/Event Hub/Function App etc.
* Technical knowledge SQL/ T-SQL/Python
* Ability to develop automation scripts

## RIDA (Risks, Issues, Dependency & Assumptions)

Testing related risks and issues will be managed by existing process within BRIT

## Reporting & Statistics

There are 2 types of test reports that will be generated throughout the delivery of testing.

* Test Progress Reports
* Test Completion Reports

All delivery/test teams across all the test phases will be expected to provide ongoing status reports tracking test/defects progress which will be centrally collated by the MDP QA team and reviewed to ensure the use case/project in on plan and quality gates will be met. A Test Completion Report must be produced at the end of each test phase to demonstrate completion of the test phase as per exit criteria.

### Test Progress Reporting

All testing metrics will be extracted from Microsoft Azure DevOps. Reports will be available via the LTE.

Reports will use the metrics provided plus additional wording to confirm any particular risks or issues which may delay current or future testing or impact quality.

Depending on the nature and size of the project the reports will include below metrics on: -

* Requirement coverage
* Test preparation progress
* Test execution progress (Pass/Fail etc.)
* Defect trends

The LTE/TM will provide a verbal update on status at regular project team meetings.  Whilst the TE will give a status update at the daily stand up.

***A dashboard can be created in ADO for each project/use case.***

### Test Completion Reporting

Test summary/completion reports will be completed as part of the exit criteria from DEV, SIT, UAT, PRE and Performance testing. These reports will confirm that all activities have completed as expected based on this document and the project test plan together with providing confirmation of the final metrics for the testing carried out.

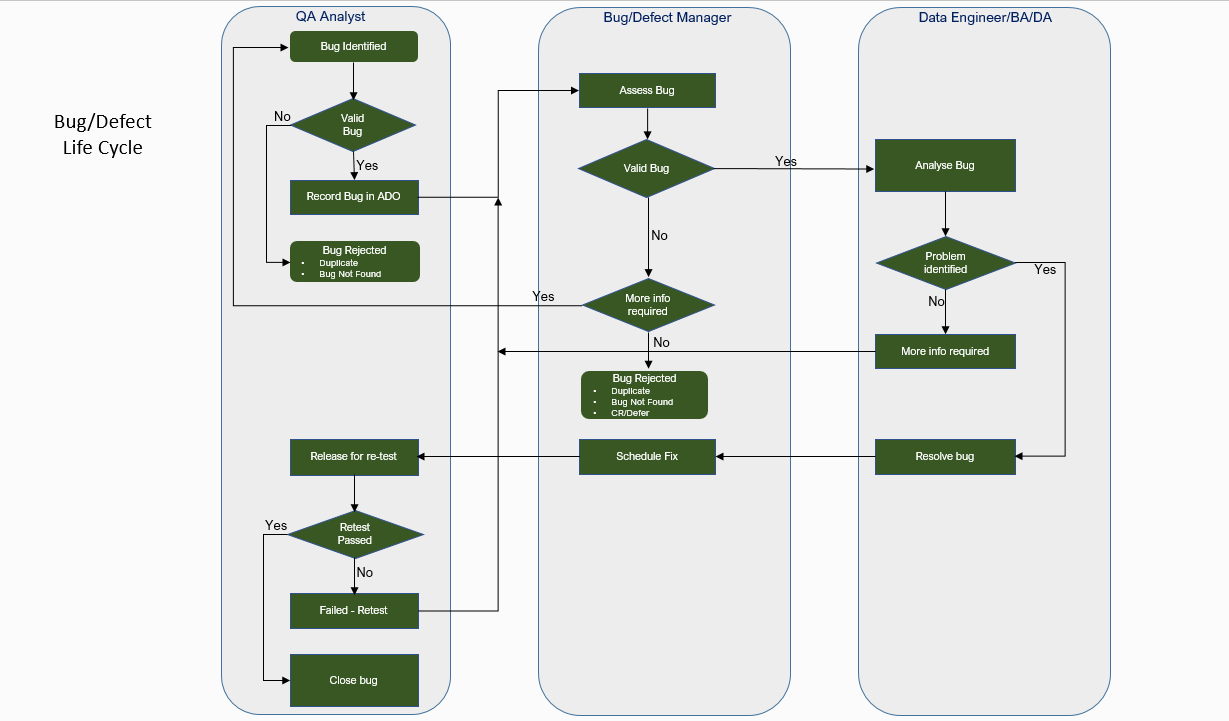
Templates can be used as per the existing processes in place (if available)

# Glossary

| Abbreviation | Meaning |
| --- | --- |
| STLC | Software Test Life Cycle |
| MDP | Modern Data Platform |
| QA | Quality Assurance |
| BI | Business Intelligence |
| ETL | Extract Transform Load |
| NFT | Non-Functional Testing |
| OAT | Operational Acceptance Testing |
| RFO | Ready for Operation |
| SIT | System Integration Test |
| PRE | Pre-Production |
| UAT | User Acceptance Testing |
| RIDA | Risks, Issues, Dependencies & Assumptions log |
| SSIS | SQL Server Integration Services |
| SSRS | SQL Server Reporting Services |
| SME | Subject Matter Expert |
| TCR | Test Completion Report |
| SQL | Structured Query Language |
| TM | Test Manager |
| LTE | Lead Data Test Engineer |
| TE/TA | Data Test Engineer/Test Analyst |
| DA | Data Analyst |
| BA | Business Analyst |
| DE | Data Engineer |
| ADO | Azure DevOps |
| LDM | Logical Data Model |
| PDM | Physical Data Model |

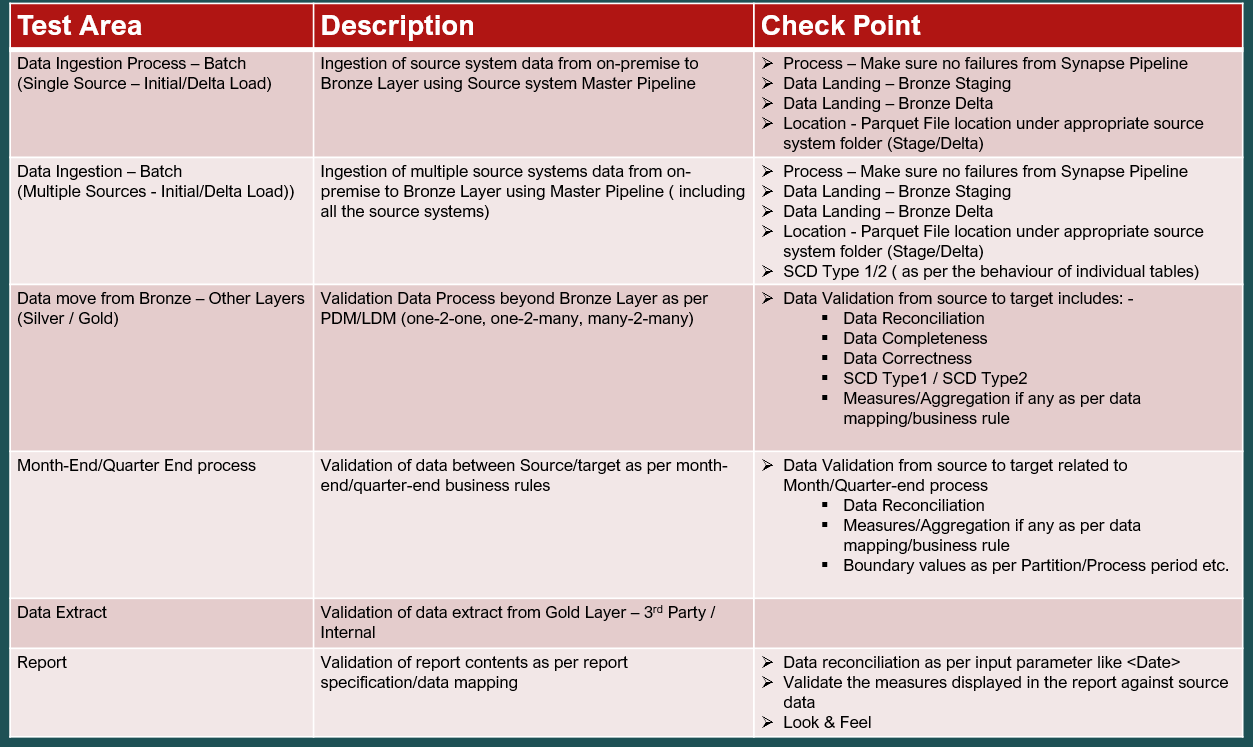
# Appendix

## Appendix A – Defect/Bug Life Cycle



## Appendix B – Sample Check Point

This is just a sample format of check point, and not related to any of the use case/project



End of the Document