Logo, company name

Description automatically generated

# 

CLOUD DATA PLATFORM – TEST STRATEGY

V0.4

VERSION CONTROL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Comment | Changes Marked? |
| Draft | 12/07/2021 | Ram Baskar | First Draft version for review |  |
| 0.3 | 16/07/2021 | Ram Baskar | Updated comments after initial review |  |
| 0.4 | 20/07/2021 | Ram Baskar | Updated comments after review |  |
| 0.5 | 10/08/2021 | Ram Baskar | Updated with Environment details |  |
|  |  |  |  |  |
|  |  |  |  |  |

Approval

|  |  |  |
| --- | --- | --- |
| Action | People | Date Completed |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

DISTRIBUTION

|  |  |
| --- | --- |
| Name | Role |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Contents**

[1](#_Toc79581672)

[1 Introduction 4](#_Toc79581673)

[1.1 Background 4](#_Toc79581674)

[1.2 Purpose 4](#_Toc79581675)

[1.3 Context Diagram - CDP Environment 5](#_Toc79581676)

[2 Project Scope 6](#_Toc79581677)

[2.1 In-Scope 6](#_Toc79581678)

[2.2 Out of scope (To be discussed) 6](#_Toc79581679)

[3 Test Approach 7](#_Toc79581680)

[3.1 Agile Methodology 7](#_Toc79581681)

[3.2 Approach and Techniques 7](#_Toc79581682)

[3.3 CDP Platform – How we test 8](#_Toc79581683)

[**3.3.1** **Data Source** 8](#_Toc79581684)

[**3.3.2** **Raw – Conformed** 8](#_Toc79581685)

[**3.3.3** **Conformed – Azure SQL DB (Staging Layer)** 8](#_Toc79581686)

[**3.3.4** **Destination (Azure SQL DB Staging – Extract – DW layer)** 8](#_Toc79581687)

[**3.3.5** **Consumers – via portal / visualisation tool / subscription** 9](#_Toc79581688)

[4 Automation 9](#_Toc79581689)

[5 Environments 9](#_Toc79581690)

[6 Test Data 10](#_Toc79581691)

[6.1 Data Requirements 10](#_Toc79581692)

[7 Tools 10](#_Toc79581693)

[8 RAIDs 10](#_Toc79581694)

[8.1 Risks & Issues 10](#_Toc79581695)

[9 Defect Management 10](#_Toc79581696)

[10 Appendices 11](#_Toc79581697)

[10.1 Digital Light House Use case – Data flow in details (Source to Target) 11](#_Toc79581698)

[10.2 Check Points – Digital Light House Use Case 12](#_Toc79581699)

[10.3 CDP/Digital Light House Environment 12](#_Toc79581700)

# Introduction

## Background

The Cloud Data Platform (CDP) is an Azure hosted data repository managed by the Cloud Data Platform Team in GTC. It aims to integrate the data from different data sources available in on-premise databases/applications/3rd party sources (like google etc.) and provides consistent information to those who need it, for use in analytics, reporting and data processing activities. It enables GTC to provide underpinning services such as retention management, data lineage and governance.

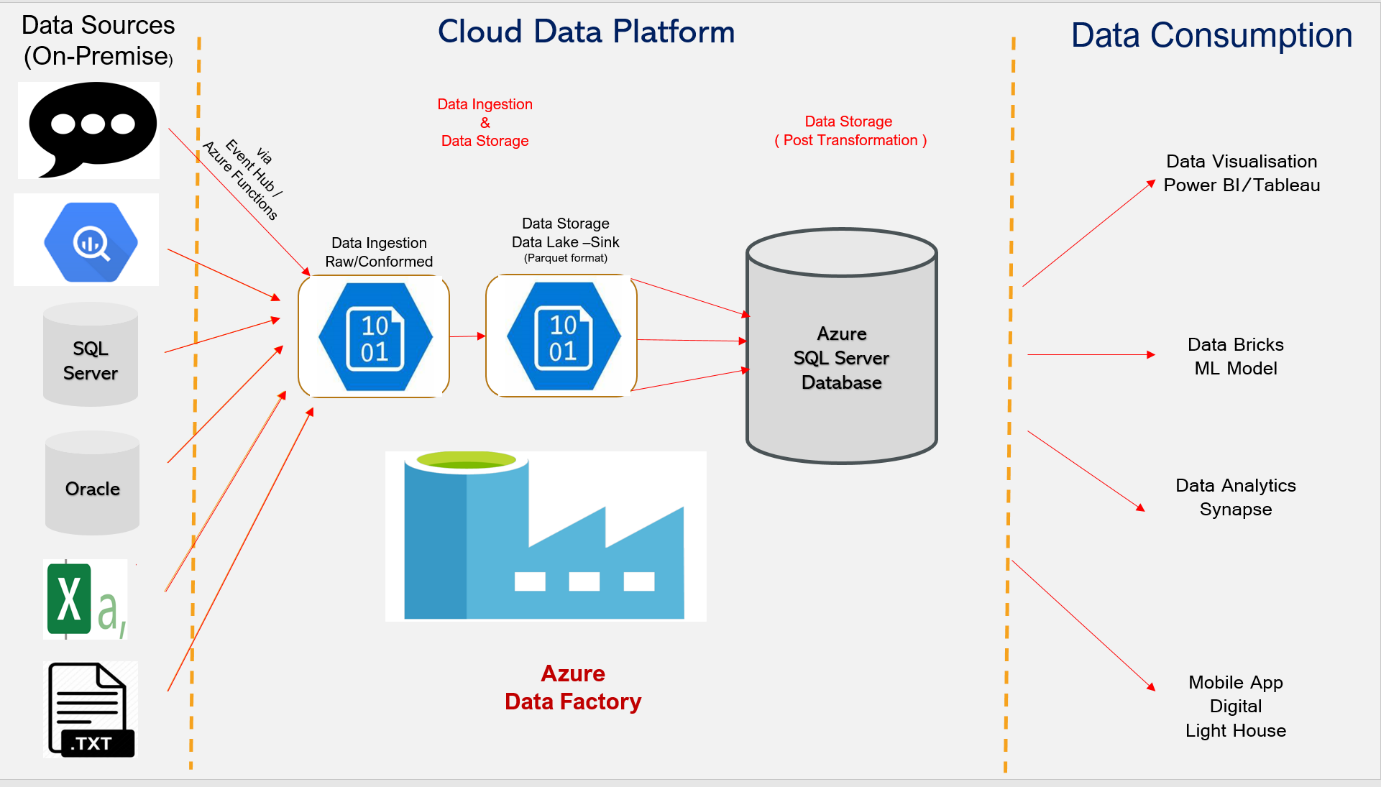
It is intended to provide all the project team members with an overview of the testing processes that will be adopted for Royal London Cloud Data Platform. This document does not intend to comprehensively detail what the coverage of testing will be for each project/use case, separate test plan will be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

## Purpose

The purpose of this document is to define how the common quality/acceptance criteria is accomplished to ensure the final delivery of the transformed data in the CDP cloud environment is fit for purpose for consumers to access the data via portal/mobile devices/visualisation tools like (Tableau/Power BI) etc.

* Define test objectives, In-scope/out of scope functions and corresponding test items to clearly understand what needs to be tested by who
* Detail the types of testing e.g. SIT, UAT, NFT etc. that will be conducted to ensure clarity on how relevant test phases will be tested
* Define an overall testing process that ensures how the whole Cloud Data Platform is tested to the degree required for consumers/business to accept the delivered system into services
* Outline how the test approach will be followed to ensure all parties understand the testing stages and corresponding timelines
* Lists the environment(s) and test tools in consideration
* Documents the clarity on the roles and responsibilities throughout the testing stages
* Document defect management process

## Context Diagram - CDP Environment



# Project Scope

## In-Scope

The following items are identified within the scope of the CDP QA team:

|  |  |
| --- | --- |
| **Area** | **Description** |
| Data Extraction & Ingestion | Extraction & Ingestion of data available in on-premises database/file (CSV/TEXT/XML/JASON etc) and 3rd party source like google big query into CDP Platform |
| EDM | Apply Enterprise Data Model schema to raw data including curation process (wherever applicable) |
| Azure SQL database | Load and apply data transformation rules as defined in data mapping |
| Outgoing Message | Sending out the processed data as agreed format to consumers as per subscription |
| Access Rights/Security  (application/data only) | Define & implement the appropriate access rights on data for consumers |
| Non-Functional Testing | Performance of all components within CDP Platform like   * Azure Data Factory Data Pipeline * Event Hub / Azure function * T-SQL process within Azure SQL DB instance * Monitoring * Failover (High Availability) * Alert |

## Out of scope (To be discussed)

The following items are identified as out of scope for CDP QA team:

|  |  |
| --- | --- |
| **Area** | **Description/Justification** |
| Source Data Quality | The corresponding source system is responsible to arrange the source data and data quality assurance |
| Security - Infrastructure | Royal London security team is responsible for infrastructure related security |
| Front-end applications | There is no requirement for front-end/web related testing in CDP |
| Latency | Latency outside CDP platform will be tested either by dedicated NFT team or by consumers |
| Downstream Applications | Applications outside CDP Platform like portal etc. Analytics related like ML Module/Synapse will be tested by data scientist in production |

# Test Approach

## Agile Methodology

CDP project will use the Agile project management methodology for Unit and SIT. The UAT/E2E acceptance and Pre-Prod approach will subject to further discussion and may undergo further changes in ways of working.

CDP QA team will actively contribute in sprint planning, analysis, and estimation of work via user story/product back log item.

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

* Walkthrough of requirements/functionality between Dev/BA to QA team via 3-amigos call
* Based on the acceptance criteria, QA team will perform the test preparation including test script/test case design/test data requirement while the feature is being developed
* Test cases will be reviewed internally to ensure that test coverage is as complete as possible and get tests ready for execution
* Once the code is ready for SIT, facilitate the team to work collaboratively to perform testing and identify the defects (if any) in the early stages
* Early involvement of business stakeholders to test specific scenarios within SIT phase
* Aim to have SIT testing completed and sign-off prior to promote the code to next phase of testing UAT/E2E acceptance.

## Approach and Techniques

To facilitate the coverage of requirements, functional tests will be executed with the multiple data sets/multiple cycles (Cycle1 & Cycle2) from source to make sure that, all the relevant scenarios are covered

Test execution during SIT phase will be carried out with at least 2 different data sets.

* Initial Load for Reference data (if applicable)
* Initial Load for transactional data
* Delta/Incremental data load – Reference data
* Delta/Incremental data load – Transactional data

Cycle – 1 will cover testing the functionality based on the initial data load. Up on completion, data in the target tables will be backed up prior to start loading Incremental data.

Cycle -2 will cover testing the functionality based on incremental data load. It also covers any fixes for the defect raised from cycle – 1.

At the end of the SIT phase, the regression suite will be built and maintained for future test execution.

For functional testing, multiple check points will be defined and verified at each step where the data is moved from source(on-premises) to cloud environment.

An example for check points for a specific use case with various stages can be found in the Appendices.

## CDP Platform – How we test

Testing CDP is different from traditional application testing as it requires a data driven approach. The data extracted from the on-premise source database/files etc. will be moving through various stages before the data consumed by different business stakeholders. The standard approach in CDP is to make sure that, the expected data from the source based on the requirement analysis is extracted and processed through various stages and is transformed into cloud environment and to ensure that, transformed data is fit for the purpose for consumers to access the data using portal/mobile devices/visualisation tools like Tableau/Power BI etc.

### **Data Source**

Identify the source data from on-premise/3rd party database/application and make sure that, connectivity is in place from ADF to pull the data into cloud environment. Depending on the source data format, this may involve running a piece of code or the data can be picked up directly by ADF data pipeline using the integration runtime. Run the ADF data pipeline “to raw container” manually. Once the process is completed successfully, verify the data is landed in data lake “raw” container with schema details as same as in source system

*Source – On-premise*

*Target – Data Lake (raw container)*

*Format – Parquet*

*Dependency – Availability of on-premise source data*

### **Raw – Conformed**

Run the ADF data pipeline “to conformed container”. Once the process is completed successfully, verify the data is landed in data lake “conformed” container with the schema details as per enterprise data model (EDM)

*Source Raw container*

*Target – Data Lake (conformed container)*

*Format – Parquet*

*Dependency – Availability of data in Raw container*

### **Conformed – Azure SQL DB (Staging Layer)**

Run the ADF data pipeline “to staging” manually. Once the process is completed successfully, verify the data is landed in Azure SQL DB staging layer with the schema details as per enterprise data model (EDM)

*Source – Conformed container*

*Target – Azure SQL DB (Staging Layer)*

*Dependency – Availability of data in conformed container*

### **Destination (Azure SQL DB Staging – Extract – DW layer)**

Run the SQL agent/T-SQL script manually. Once the process is completed successfully, verify the data is landed in “Extract” and “DW” layer. Verify the data transformation / (aggregation if any) at destination layer(s) as per transformation rules defined in the data mapping interface. In addition to this, other tests including data integrity, data completeness and schema validation will be carried out

*Source – Staging*

*Target – Extract & DW layer*

*Dependency – Availability of data in staging layer*

### **Consumers – via portal / visualisation tool / subscription**

Consumers are able to access the data either by direct access via security implementation like RBAC etc. to ensure that data is accessible only by authorized users and are available only to them or CDP sends out data in an agreed format like XML etc. based on subscription via event hub

# Automation

The automation process within CDP will be a work-in-progress activity and the key area to automate is as below. To start with manual~~/semi-automation~~, each of the below area will be automated (where possible) to achieve the benefit of running the tests quickly with the aiming to achieve 100% automation.

|  |  |
| --- | --- |
| Test Automation Area | Description |
| Azure SQL DB | Validation - Data Mapping / Data Integrity / Transformation |
| ADF - Pipeline | Process to extract & load source data |
| Azure Functions | Process to extract & load data via Event Hub |
| Outgoing Message | Outgoing data CDP - Consumers based on subscription |

# Environments

Summary of environments as below (Being discussed)

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

# Test Data

## Data Requirements

One of the main deliverables to start SIT is the source data and will be sourced from the relevant source system(s) involved in the individual use case as per the agreed interface between CDP/Source system owner including column level details with good quality. CDP QA team will request test data either cut down version or full copy of test data (based on the individual use case requirement) for an initial load, followed by an updated data for an incremental/delta load. CDP test team is responsible for extracting and processing the data across the multiple layers and eventually transformed and loaded into Azure SQL DB. CDP QA team is also responsible for validating data reconciliation between Source/Staging layer, data transformation and data integrity across the objects/tables. During E2E/UAT acceptance testing, meetings will be set up with the relevant source system owners to understand the data requirements to cover the agreed scenario(s). However, an early involvement of other stakeholders will help to find the issues (if any) in early stages by testing business critical scenarios as part of SIT phase.

# Tools

|  |  |  |
| --- | --- | --- |
| **Tools** | **Name** | **Purpose** |
|  | Azure Portal ADF / Data Lake) | **ADF** – Azure Data Factory is the tool used to extract/ingest data into the CDP platform using data pipeline.  Copy activity is one of the tasks carried out within ADF using **Linked Services** by defining **source/sink** details  **Data Lake** – Holding the storage account with multiple containers to store the data after copy data activity from the source |
|  | Azure SQL DB | Cloud SQL server database used to store the data after transformation from where will be consumed by consumers |
|  | PowerShell | Utility to run the commands in shell and will be used for automation using pester framework |
|  | T-SQL | Programming language in Azure SQL server database – used for data move from staging to target schema including transformation |
|  | Data Bricks | Tool which can be used for analytics purpose to manipulate data using wither Spark-SQL / Python etc. |
|  | Test Management - ADO | Tool where the tests/test cases will be created and managed for reporting |
|  | tSQLt | Unit testing Framework used by Data engineers |

## 

# RAIDs

## Risks & Issues

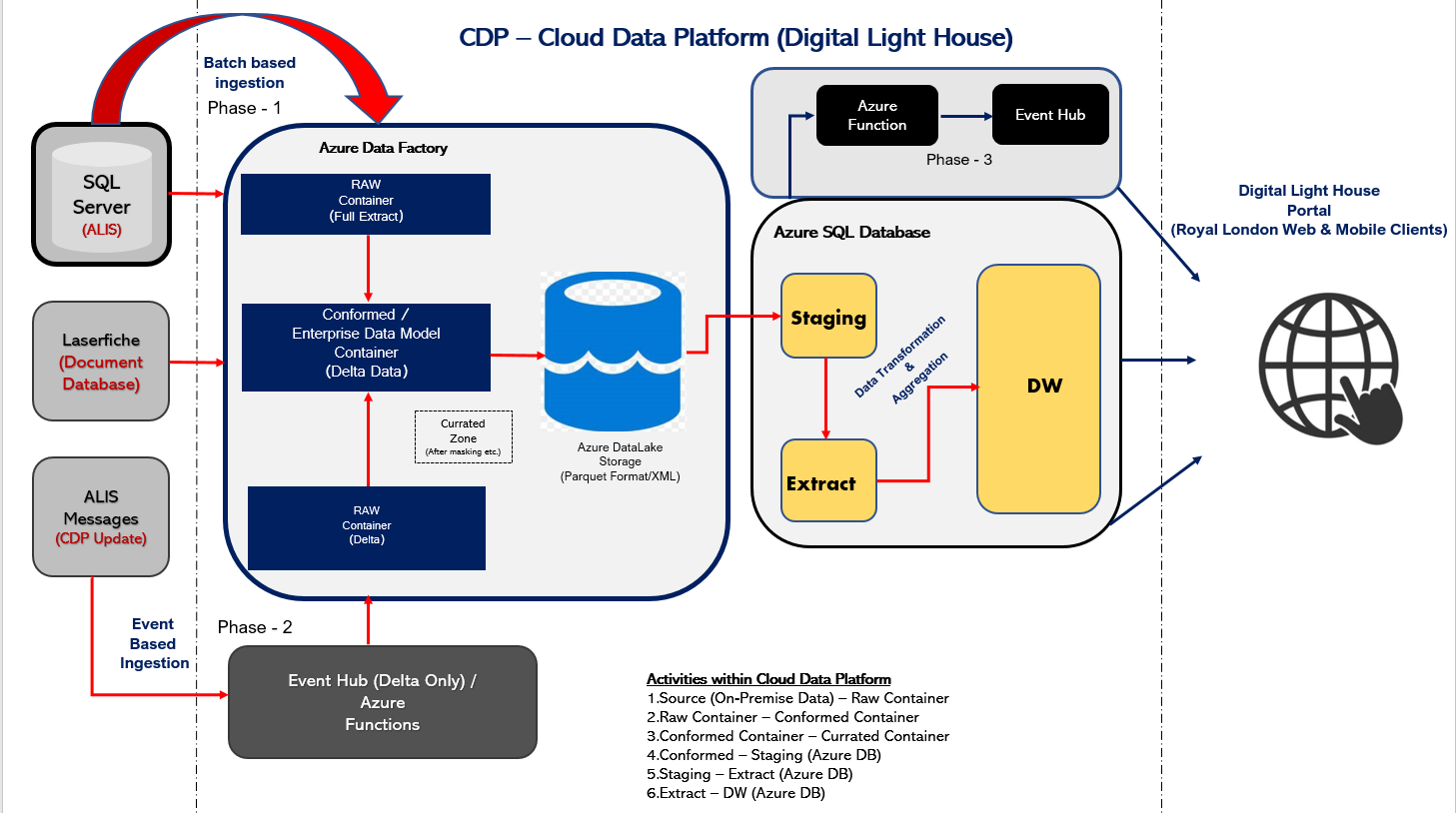
Risks and issues will be recorded on the project risk log as per Royal London standard process.

# Defect Management

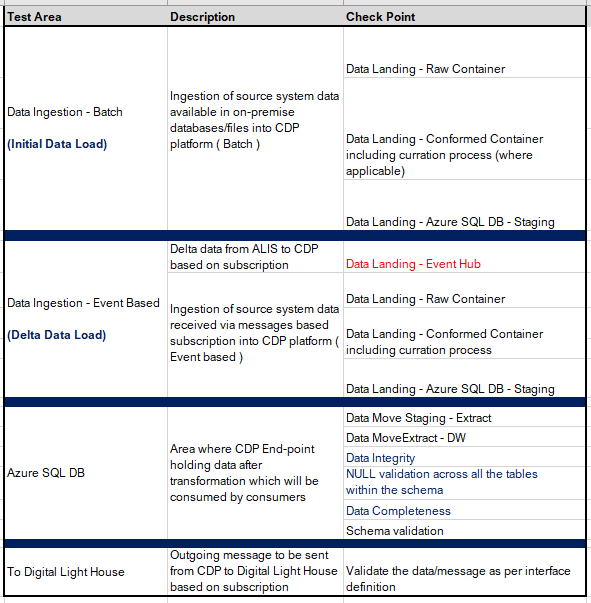
Standard Royal London process will be followed for defect management

# Appendices

## Digital Light House Use case – Data flow in details (Source to Target)



## Check Points – Digital Light House Use Case



## CDP/Digital Light House Environment

|  |  |  |
| --- | --- | --- |
| **Environment**  **(CDP)** | **Environment**  **(ALIS/Protection)** | **Description** |
| TEST | TSA | SIT Functional + Security (application/data) + Component Integration (within CDP) + NFT |
| UAT | PPR | Integration Testing (CDP + Portal/Visualisation Power BI/Tableau etc.) |
| Pre-Prod | TSP | Non-Functional Test based on NFT requirements |
| Prod |  | RFO – Ready for Operation (Data Reconciliation) |

**CONSOLIDATED FEEDBACK FROM TEST QUALITY MANAGEMENT TEAM**

|  |  |
| --- | --- |
| **FROM** | **FEEDBACK** |
| Luca Ramires | Document is not up to RLG standards yet, several sections seem to be missing.  Not clear if it is a Strategy or an Approach (the two docs have different scope, really).  **(A) If you consider it a Strategy**, I’d like to suggest some good examples to consult and follow a similar structure (they are not in Standards Book yet but they will soon):  A1) Protection Portfolio Test Strategy    As regards with Automation and NFT, please feel free to consult the structure of these two documents:  A2) Protection Automation Strategy    A3) Protection NFT Strategy    (…you could join some of the sections of three documents into one document, if needed).    **(B) If you consider it an Agile Test Approach**, please use the Agile template approach v1.0 in [Standards Book](http://developmentpracticesharedprocesses:43211/gtcstandards/default.aspx).  **(C)** **Additionally, starting the analysis from the** Architecture Data Strategy (created by Owen Jones and approved recently) could be helpful to share the same language and principles (i.e. what democratization of data really means for RLG): |
| Stephen Worlock | Firstly, we need to accept that this is going to be a difficult Strategy to pull together as the language, processes, technology and ultimately risk are unfamiliar. I suggest ‘small steps’ in that the 1st draft of the strategy should focus on:   * Who is needed to compile a cohesive strategy and by what mechanism (i.e. series of workshops) * The key business risks that we are trying to mitigate with our testing * Analyse the profile of the testing you could do and sketch out the testing in terms of   + a) what will QA need to control data in to a test   + b) what will QA need to run the test through the solution   + c) what will QA need/use to observe the various facets of the result   + d) what will QA need to compare those results with what is required. * introducing Testing terminology the programme will use and stick too (solution architecture as input)   As required, the governance aspects – forums, deliverables, entry / exit criteria, signoff protocols and all the other standard Test Strategy sections can be added in a later draft. |
| Dhuruvan Ekambaram | * Overall strategy does not fit to the org template/standards and is inconsistent with rest of the approved strategies signed off Portfolio Strategies RLG (happy to share available templates). * The overall document needs to be an e2e document covering all aspects of testing but does not cover business testing, UAT and acceptance criteria * Do we need to add a glossary? * What is scope for accessibility testing? * What are the test deliverables produced and delivered here? * What sort of test metrics and progress reports will be considered? * Entry and Exit criteria for testing is missing * Suspension /Resumption criteria for testing is missing * Some example tools to consider for automation: Segment, Tealium & Optimizely, also does not Cognizant offered Platinum tool be used here? * It appears the strategy is at draft level however it must evidence what aspects are covered for testing and how we are proving our exit criteria with other downstream dependent projects (i.e. Digital Lighthouse) a demarcation of how defects will be handled etc needs to be agreed. |
| Prasanna Raja | * It was captured the details (Unit Testing, SIT, etc) at very high level but when we talk about the Data testing, it should cover some test techniques / types in detail – example, migration testing, reconciliation testing, interface testing, etc. (see example below in **purple** from Data team in Legacy portfolio), without that it would be a bit difficult to call it a ‘test strategy’ for the data testing projects. Section 3.2 covers some of it, but not all / in detail. * No details around ‘Management & Quality Controls’ for release management, ‘Test Deliverables’ for the delivery, etc which are our usual standards to list out in any test strategy. Probably its worth asking them to look at any one of our Portfolios test strategy document for reference.  Test Techniques  **Data Migration Testing / Source-to-target Data Testing**  In this type of testing, a tester validates data values from the legacy (source) system to the Integro (target) system. It ensures projected data is added to the Integro system without loss or truncation, and that the data values meet expectation after transformation.  Data testing will be performed either by running the Test SQL or building a test workflow in Alteryx Tool based on the source and Target type (file/database type).  **Dependencies:** Source to Target Mapping specification, File specification, Source File decoder  **Automation Tool:** ETL Powershell tool / Alteryx    **Regression Testing (Fully Automated)**  Regression testing promotes the improvement of the product quality, and it verifies that any modifications that have been made do not impact the core functions already tested and proven.  Test SQL’s designed & executed during every sprint is stored in the master regression suite. This suite will be kept ready in specific form to be run in the ETL PowerShell test tool  **Dependencies:** Training for the testers, Additional (SQL server) connectivity in ETL PowerShell tool  **Automation Tool:** ETL PowerShell tool    **Interface feed Testing**  Purpose of the test is to verify the interface file (produced as per the defined specification.  Testing is focused on the file layout, Batch job& data elements. Alteryx workflow will be built to verify the business rule defined for the data population.  **Dependencies**: File Specification with clear criteria/rule  **Automation Tool**: Alteryx tool    **Reconciliation Testing – Unit of Migration (UoM)**  Purpose of the reconciliation is to ensure that all required legacy data has been accurately converted to the Integro system.  It is important that we manage and report on it at a UoM level. We must be able to show that the number of UoM that commenced the journey equal the number of UoM that ended the trip less any that we know fell foul of Validation, de-duplication etc. Those that did drop out en route to be accounted for, identified and made available for possible remediation. This is where we count not just the number of UoM occurrences before and after the migration but also meaningful values within them to make sure that our transformation, data enrichment, consolidations and exclusions have not altered a fundamental aspect of the UoM.  **Dependencies**: We need suitable legacy report/outbound file to compare back to  **Automation Tool**: Alterx tool will be leveraged to reconcile the data  UoM - Unit of Migration **User Acceptance Testing**  Acceptance testing is to be carried out by business users in a dedicated environment focusing on user requirements and needs. The purpose is to revalidate the outbound feed that are fit for use that work alongside the solution being delivered.   **Operational Acceptance Testing**  Operational testing in the acceptance test phase will be performed in a (simulated) operational environment by operations and/or systems administration staff focusing on operational aspects, i.e. recoverability, resource-behavior, install ability and technical compliance **Non-Functional Testing**  Performance testing to be carried out by NFT team. Separate NFT test plan, scope will be drafted based on the Project requirements. |
| John White | * I don’t fully understand what the CDP will be providing. Is it purely to identify opportunities? With the being erroneous data captured and migrated over the years, how will the impact of this data be managed. |
| Ed Murphy | * It isn’t a strategy document. There’s no “why” detailed to the testing, e.g. data consistency/completeness/accuracy * If it’s an approach, then just use the Agile Test Approach template [https://royallondon.sharepoint.com/sites/GroupTechnologyandChange/\_layouts/15/Doc.aspx?OR=teams&action=edit&sourcedoc={D6C4D386-32ED-4D07-96D5-5A222E92DFBB}](https://royallondon.sharepoint.com/sites/GroupTechnologyandChange/_layouts/15/Doc.aspx?OR=teams&action=edit&sourcedoc=%7bD6C4D386-32ED-4D07-96D5-5A222E92DFBB%7d) |