**Smoke Testing Cycle 1&2 – Summary Report**

**Mainframe Integration Behavior Overview**

Smoke tests were run over valid and invalid test scenarios on behalf of the mainframe COBOL modules during this testing cycle. These tests were mainly to verify core application functionalities, system responsiveness and integration reliability prior to launching into regression or full validation test.

**Key Observations**

**1. Valid Test Cases Execution**

* Test cases that pass valid data into all critical modules under test, such as report generation, data entry, to batch processing, passed successfully.
* The test cases for valid data actually ran successfully interacting with the Oracle database in COBOL programs and mainframe output matched the expected result.
* Execution times of standard report modules met acceptable run times.
* Batch programs of true sequence without interruption to accurately reflect the data at the store level.

**2. Invalid Test Cases Execution**

* In every test inputs, invalid data (e.g., malformed data entries, boundary violations, etc.) and unauthorized operation always trigger appropriate COBOL error handling routines.
* For each failure, ABEND codes were captured in system error logs, so that there was traceability in the logs.
* Mainframe backend response codes were defined to provide oracle frontend behavior in terms of these codes preventing data corruption or inconsistent data.
* Legacy COBOL modules were occasionally without user friendly messages, thus suggesting the requirement for standardized error responses.

**Challenges Identified**

**Data Mapping Inconsistencies**

Consistent with invalid cases, discrepancies were found between data displayed in Oracle and COBOL generated records. Such may arise of a mismatch between field length or type when sending data.

**Error Propagation**

Some of these invalid test scenarios had failure cascades in subsequent downstream jobs because batch error management was not isolated between jobs, especially during concurrent multi-store data simulations.

**Execution Delays**

In I/O intensive COBOL Schedules, the job completion is delayed using larger test datasets Albeit malicious using real time volumes.

**Suggestions & Recommendations**

1. **Enhanced Mock Data Usage**

Mock data framework is implemented to simulate different valid and invalid conditions without risking production like data.

1. **Early Integration Simulation**

Perform the Oracle-to-COBOL data format transition earlier in the SDLC especially, and focus during this integration testing on the problems in Step 1**.**

1. **Error Message Standardization**

Design a new module to handle update COBOL errors to provide clearer and consistent messages across modules.

1. **Parallel Execution Profiling**

For large volume batches, ensure batch program threading and scheduling do not overload mainframe job queues.