Technology Migration at KGFS
- By the IFMR Rural Finance Technology Team.

Recently we migrated from our legacy system to a new core banking system. This note describes the activities involved in migration, the methodology to be followed, step-wise strategy and requisite precautions that need to be taken pre and post migration

The core of all financial transactions is handled through a core banking system. Typically all asset products (loans, overdrafts), and liability products are hosted in a core banking system. The system records transactions, maintains customer information and calculates interest on loans and deposits etc. The same software can be accessed from various branches of a financial institution.

There are three reasons for migration:

1. Change of system (service provider)
2. Upgradation of application
3. Upgradation of technology

Migration is of two types:

- Migration from legacy system to new system
- Migration of data to existing system

Multiple systems have their own way of performing calculations and storing data. Systems may provide the same functionality but differ in architecture. Difference in data structure and architecture makes the migration process a complex one. All information available in the existing system(s) may not be directly usable in new system. The data from the legacy system has to be modified/processed before loading in the new system.

Steps followed during migration:

1. Identifying data requirements for new system
This process begins with identifying data required by the new system.

- Pre-migration activities – This primarily consists of understanding the data structure of the legacy and the new system, one to one mapping of fields in legacy system and new system, identifying and rectifying gaps, and collating the data in a form which can be transferred to the new system. This stage encompasses mock tests/dry runs to check the correctness of data and modifications if necessitated.

- Post migration testing/consistency checks – This includes checking the data sanctity after it is moved to new system, checking accruals, payoffs and other calculations which have impact on the business. This is done to validate for accuracy, precision and completeness of data followed by full scale migration. After testing is completed, a final acceptance/roll-out testing is performed to certify that the data migrated confirms to the requirement.

Generally for asset products, data requirement can be classified in three categories:

- Product features information – This consists of creating products with features in which the accounts in legacy system are getting serviced.
- Instalment information – Information pertaining to loan records and bill related information is inserted.
- Loan History information – Transactional history for all the accounts are uploaded.

2. **Format of data files to be uploaded**
   Format of data is mutually agreed upon by both client and partner. Data format requirement is more or less driven by system architecture or database design of the new system.

3. **Confirmation of data sent and data uploads**
   Once the data format setup is done, files as per agreed formats are uploaded in testing server. Validations using built in utility check for data is performed before data is finally uploaded to testing server. Validation utility provides confirmation on number of records uploaded and rejected. Based on this information, user can correct the data format.

4. **Testing of all the scenarios**
   Once the data has been uploaded to the testing server, all stakeholders test the various scenarios of the lifecycle of loans and share observations. These scenarios are also known as test cases.

5. **Issue resolution (iterative process)**
   The Implementation team identifies the failed test cases and resolves them. This is an iterative process till all issues are resolved.

6. **Mock Migration**
   Once all the issues are resolved in testing, one should plan for mock migration. Mock migration means arranging a separate migration server and copying the same production application code along with database. Live database migrated to migration database should be masked to ensure confidentiality of information. Once production copy of both application and database is migrated to migration server, the migration data is uploaded once again. Post upload, test cases are validated once again to ensure that all the data has been migrated successfully and accurately.

7. **Actual Migration**
   On successful mock migration, actual migration can be initiated. Ideally, actual migration is planned on holidays to have sufficient time for system to recover in case of any issues and to not let business operations suffer. Migration checklist should be checked once again to ensure that all requirements and environments are set. Once checked, migration process can be started. Migration time is dependent upon the number of accounts to be migrated.

8. **Handholding and close observation on system post migration**
   Utmost care is taken in the initial days post migration to ensure that data processing is happening as expected and EODs are happening in the expected time.
   The approach adopted was all branches and business going live simultaneously. Although this results in faster implementation it calls for a huge onus on the part of resourcing. The error tolerance level is also low in this case as all the branches and all the accounts are migrated simultaneously. This Approach is also called the “big-bang approach”.

Takeaways from Migration

- Understanding of data structure and account behaviour in legacy and new system is a key to successful migration.
- It is advisable to always plan for mock drill before actual migration. This has to be accompanied by thorough testing and verification of data post migration.
- Training should be given to users and they should be educated on account behaviour in the new system for accounts coming from the legacy system.
- Have patience.