



Thumb Rules

of Data Migration

By

Jayashree Patil

Senior Data Architect, Tavant



Content

1. Requirement Analysis
2. Detailed Mapping Design Specification
3. Understanding Business Rules
4. Setting Up the System Configurations/Parameters as per the New System
5. Technical Design
6. Migrating the Data Without Impacting the Existing System Performance
7. Testing and Verification of the Migrated Data
8. Data Migration Summary Report
9. Performance Testing
10. Data Migration Project Planning and Constant Interaction with Business Owners

1. Requirement Analysis

Requirement analysis helps in understanding both the source and target system in terms of business process, application functionality, data mapping, extraction logic, the gap between two systems, understand the scope of the migration, and many more.

Outcomes:

- Number of source systems/applications, databases, and interfaces to migrate
- Lists modules/subject areas those are needed to be covered as a part of migration (gives scope for data field mapping)
- Number of integrations to be covered
- Provide details like mode of access to the source systems, source data extraction window period, and others
- Helps in understanding the relationship between the entities (ER model) for both the systems (source and target)
- Lists data challenges like missing data, date format issues, dissimilar data structures for the same customers, products, and so on
- Approach towards protecting sensitive data, if any (data encryption needed for specific data attributes like customer information)
- Data volume for each entity would help in identifying the data size, and data movement approach especially for (BLOB, CLOB) data types
- List out incremental migration steps wherever needed
- Number of business rules needed to remap or recreate in the new system
- Gives downtime requirement details for both the systems during go-live
- Helps in finding the right data owners for data reconciliation process towards each module
- Provides better estimation or timeline needed for project execution, risk or challenges to address, and more
- List actions needed for data archival, if applicable
- Lists all the known logics/methods that get implemented at the application level and which will have an impact on the data too. Examples based on different criteria and conditions are applied to the objects.
- Lists users access details based on the new system in comparison with old systems
- Helps in listing all the test scenarios based on each entity and across entities/attributes
- Helps in evaluation and implementation of new application features to support legacy data/features
- Estimation of hardware and software requirements
- Determines the transition process to be followed

Note: All the data quality issues will not be identified at the requirement analysis stage.

Why is it important?

Requirement analysis provides what is needed to be implemented or expected to do, what are source system, and each source system's implications on the new system, etc. The requirement analysis contains the behavior, attributes, and properties of the old and new systems. Therefore, the main task of the requirement analysis is to ensure that all stakeholders are aligned with what is needed to be done and the expected result.

How it impacts?

If we do not understand what is needed to be done, it will be a trial method and leads to a lot of rework. So, the result of the failure is significantly high (up to 80%).

How to make it more effective (Recommendation)?

Requirement analysis should be as detail as possible to make a proper plan for data migration tasks. Detailing out points mentioned in the "outcome" is a must and keep the checkmark for each point as per the project need.

2. Detailed Mapping Design Specification

This process will help in finding a thorough specification of how the source and the target entities will be mapped down to the attribute level. Mapping gives a clear understanding of attribute details like data type, data length, date formatting, transformations/derived attribute details, default values for certain mandatory attributes, referential integrity constraint details, splitting, merging of the data, and so on.

Outcomes:

- The mapping document will have details of data transformations for each entity along with joins, derivations, default values, date format, etc. It will be a crucial document for data extraction and loading.
- It also provides details on what order to follow for data extraction (data extraction methods), data relationship, data dependencies list, data transformation rules, and others.
- The mapping document will act as metadata that can be further classified/segregated as business and technical metadata. If we use any tool to migrate data, the metadata can aid in helping automation of the jobs and take part in every data migration process.

Why is it important?

Data mapping is an essential step in the data migration process. It is a mechanism that matches fields from data sources (system A) to the target fields (system B). Data mapping is a must task to start the data migration.

How it impacts?

The mapping document is a vital task in the data migration process. If we do not do this step correctly, then the probability of data migration failure is 100%.

How to make it more effective (Recommendation)?

1. *Mapping at each attribute level and getting it verified by business with the sample data would help in avoiding remapping tasks.*
2. *If we have derived fields at the application level in the old system, those logics need to be verified with business stakeholders and should be maintained in the mapping document.*
3. *Get the sign-off on the mapping document from the business. This will help in keeping everybody (business, users, development, and testing team) on the same understanding.*

3. Understanding Business Rules

This process helps in segregating the business rules required at the application and data migration level. Certain business rules will be defined at the application level. These rules will be applicable to the new transactions only, while other business rules are defined at the data migration level. This needs to be considered during the migration process.

Outcomes:

- This process helps in identifying business rules and segregation of the regulations. Application related business rules require additional steps at the application level and will not be part of the data migration scope.
- Helps in listing in the format required to migrate the business rules as per the new system
- Lists impacted modules and their corresponding entities/attributes

Why is it important?

Business rules are used mainly to automate certain tasks/processes in the system to avoid human effort or manual processes like certain payments will be done based on business rules and criteria. So, it is important to know the business rules that impact the data to be migrated.

How it impacts?

The criticality or impact of the business rules are based on the automation process in the system if the payment module has business rules setup, then it has a major impact.

How to make it more effective (Recommendation)?

List out all the business rules and understand the impact of each business rule whether its applicable for the data migration. If yes, then how it impacts before and after data migration.

4. Setting Up the System Configurations /Parameters as per New System

Every application or system will have a set of configurations and parameters to configure at the initial stage, which may have an impact on the data that needs to be migrated. These configurations and parameters need to be considered during the data migration process. This process will help in listing out all the configurations/parameter details.

Outcomes:

- Lists all the system configurations
- Lists and redefine all the master data required as per the new system
- Lists all LOV (list of values) as per the new system
- Helps in listing 'Internalization of the data list'
- Lists all the impacted entities/attributes, which are required to incorporate as a part of the data migration process

Why is it important?

Certain system configurations and master data setup will be done to maintain data consistency, data quality, and data validations within and across the system. It is very important to understand these setups when we are migrating data from one system to another. We need to understand how to remap/reuse these with a new system and its impact on data changes.

How it impacts?

These parameters drive the system, so it is imperative to know the configuration needed for the system and what all need remapping. This might lead to an 80% failure.

How to make it more effective (Recommendation)?

List out all the business rules and understand the impact of each business rule, whether it is applicable for the data migration. If yes, then how it impacts before and after data migration.

5. Technical Design

In this process, the main task is to define the technical architecture and design of the migration process. There will be high-level and low-level design details.

Outcomes:

- A high-level design document gives an overview of the technical design of the data migration process. It contains end-to-end steps of data migration details like connectivity, extraction strategy, staging data (schema), validation process, data loading, reconciliation process, and interface details (if any) and more
- Low-level design document will have detailed steps, such as:
 - (1) The extraction process for each entity, and how many tasks or jobs needs to be created, what are all the joins and filters needed for each entity and others.
 - (2) Metadata design/architecture details
 - (3) Staging entity creation details and extraction jobs description
 - (4) Validation process/job description details
 - (5) Loading process/job description details
 - (6) The order in which all the tasks/jobs execution needs to be performed
 - (7) Contains details of the reconciliation process
 - (8) Contains details of data archival process
 - (9) Contains configuration/parameter details

Why is it important?

This document provides all the details of migration tasks that need to be performed in order. This is very important and the first document for the development team to start the work. It communicates detailed steps of how the data migration of the source and target systems needs to be performed. This helps in a sustainable development.

How it impacts?

This is a basic document needed for the development team to start their work. If this is not available, then there will not be any data migration work, and it will be 100% failure.

How to make it more effective (Recommendation)?

List all the tasks in detail, as much as possible, this will help the development team to do their work at its best and avoid a lot of rework.

6. Migrating the Data Without Impacting the Existing System Performance

As we know the volume of the data as a part of requirement analysis, we need to list out all the data movements either during off business hours or weekends and how much downtime is required during go-live, etc.

Outcomes:

- A timeline will be defined for each module/entity to be moved to a new system in consideration with dependencies and performance
- Lists all the technical steps (DDL/DML) that are needed to be taken care of during the data movement
- Consider incremental approaches based on data dependencies
- Lists all technical steps to be performed post-migration to hold better response time in the new system

Why is it important?

With agile methodologies, most of the system will be up and running while we do the legacy data migration. It is essential to consider the performance impact of the new application while migrating legacy data. This step would help in streamlining the data load window.

How it impacts?

System response time is the most significant requirement for any user, so the data migration team needs to be aware of it during the migration process. If we do not consider this step, then it might lead to the production system downtime due to data locking and blocking, etc.

How to make it more effective (Recommendation)?

Proper communication to all the business stakeholders and users is needed for the scheduled migration tasks, to make them aware of the impact even during off business hours.

7. Testing and Verification of the Migrated Data

This process provides the list of scenarios needed to be tested for each entity and ensures data consistency between the two systems.

Outcomes:

- Lists all the test scenarios at the attribute/entity level. The derived attributes/columns value will have one or more test scenarios.
- Lists all the test scenarios at business rules level
- Lists all the co-existence of scenarios between the two systems
- Defines clear test scenario on data merging, splitting or conversion
- Lists test scenarios, which are common to both migrated data and interfaces (integration/sync data). This will ensure that migrated data is in sync with interface data.
- Defines the scope of the users/customers test scenario and provides a timeline for the customer to test the migrated data (at an early stage QA environment only)
- Lists the functionality related test cases with a variety of data samples
- Provides the list to verify the changes made to support the new functionality delivered as a part of the new system
- Provides the list to verify the users from the legacy system, who can also continue to use both the old and new functionality, especially the ones where the changes are involved
- Lists all the pre-migration, migration, and post-migration test scenarios

Why is it important?

Testing is an essential part of data migration; it ensures the data movement has been done as per the requirement, data counts between the system matches and identifies the data issue or functionality issue due to the migrated data, etc. This is a must-do step.

How it impacts?

Before delivering any code or system, testing is an essential task to ensure it has been performed as per the requirement. This task has a direct impact on the success of data migration.

How to make it more effective (Recommendation)?

Communicate all the test scenarios covered for each entity or module with business stakeholders and users to ensure all their functional requirements have been addressed, and the system is ready for their use.

8. Data Migration Summary Report

Data migration summary report gives a complete detail on the data migrated at the entity level and the module level. This report provides an overall data count for the migrated data, duplicate data, and error records for each entity/module.

Outcomes:

- Defines all the details required to be covered as a part of the summary report (format of the report)
- Defines error report format and adds the data owner name for reconciliation of the error report
- Provides information on the data quality issues, data mismatch, data loss, and data volume for each record

Why is it important?

This report is critical to the developer, business users, source data owners, and management to know the data summary at a high level and provides detailed information for the development team to get the data reconciliation. This report provides complete information of the migrated data like several source records available, a few records loaded into the new system and, a few records to be reprocessed, etc.

How it impacts?

It has a direct impact on the data migration completion sign-off from the business. This report will be used by a business stakeholder to verify that the data counts across systems are same or different. If it is different, then it should provide details on the reasons for which data counts are different, etc. These summary reports will also be used for data audit purpose. Hence, it has a significant impact on the business as well to get sign-off from the data audit team.

How to make it more effective (Recommendation)?

Ensure all the data error messages are self-explanatory instead of a single word so that businesses would be able to understand and provide a quick reconciliation resolution on the data error records. Segregate or keep a separate report for the data, which will not fit in the new system, but it's required as data reference as part of the legacy system, this helps to avoid confusion or conflict on the reconciliation process.

9. Performance Testing

This is a crucial step to perform after the data migration, and it is mandatory. Performance testing is important to ensure that migration to a new system/software has not degraded the performance of the system.

Outcomes:

- Defines the criteria and test scenarios for load testing
- Also, certain scenarios for manual testing are defined, which are not covered as a part of the load testing

Why is it important?

The end user of the application would always expect better response time for their daily operations. Performance testing is done to provide business with information about their application regarding speed, stability, and scalability.

How it impacts?

As we migrate lot of data from legacy system, it is an essential step to do the performance testing, to know the business user's feedback on the system response time, and action needed if the response is not as per the business expectation. So, it has a direct impact on the business operations.

How to make it more effective (Recommendation)?

Performance testing should cover all the functional testing along with the load test on the actual size of the data (replication of the prod data).

10. Data Migration Project Planning and Constant Interaction with Business Owners

The project planning must have an estimate for all the tasks mentioned above along with timeline details, task dependencies, task owners, and more.

Why is it important?

Taking proper planning steps before initiating a data migration project can be crucial to success. A complete and accurate picture of the data migration steps, dependencies between systems, teams, etc., existing/new infrastructure including all data, hardware, software, and network components is necessary for a successful data migration plan.

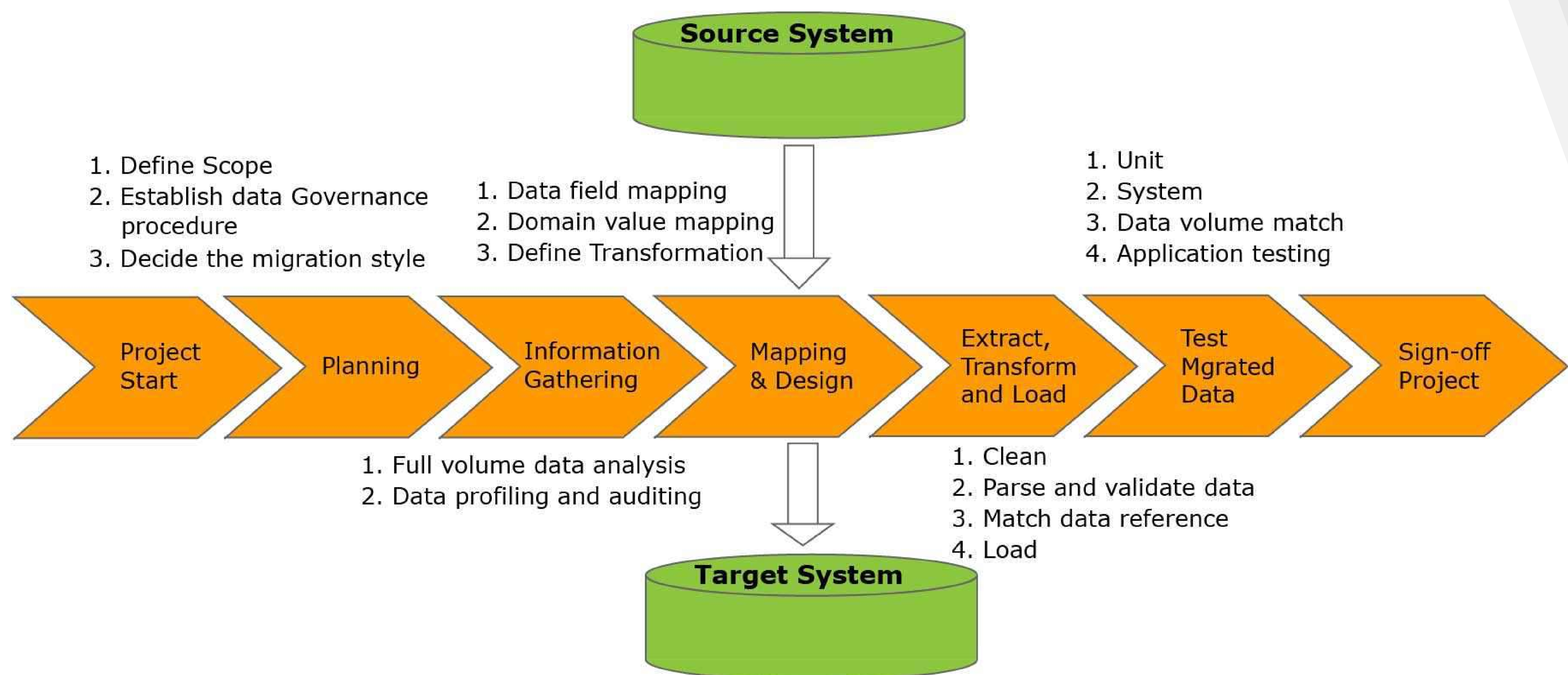
How it impacts?

Planning is a must, and it directly impacts the success of the data migration to have a smooth execution of the data migration. There is a 100% failure if not appropriately planned.

How to make it more effective (Recommendation)?

Data migration planning should cover as many details as possible, along with all the dependencies and certain additional buffer for the data reconciliation process. Since, some data issues and data change requirements will not be known until the migration is done.

The successful data migration project will have all the tasks mentioned below in diagram:



Reasons for Not Meeting the Timeline:

The main reason for data migration projects to delay or not to meet the requirement of the migration can be due to the following:

- The requirement may not be precise, implying the data migration project is not discovered well.
- Wrong assumptions, based on the previous experience without knowing the actual requirement of the source system
- Data quality issues may not be appropriately analyzed, and the owner of the data correction or data reconciliation may not be clear. The data reconciliation is a big task in data migration, if the quality of the data is bad. So, we need to find the right owner of the data to make proper data corrections, if required.
- Not having proper documentation on the data flow, which will take a lot of time to understand the legacy system
- Not having all the required details on carrying out proper project planning such as efforts applied on a task, impact of the dependencies across the teams, and the systems analysis
- The development team might not get a clear understanding of both the systems. No proper training on the old and new system functionality, data flow, etc., will lead to functional issues, causing a rework on the same data repeatedly.
- If a new system comes under the development phase and the data migration starts parallelly, then there could be a lot of rework. In case any further changes are made to any module in the application, it is based on customer feedback or to incorporate certain functionality.

Outcomes:

All the stages in the thumb rules defined above will apply to every data migration project. But the duration and the activities performed in every phase will depend on the size and requirement of the data migration process. The success of the migration will entirely depend upon how well we follow these ten thumb rules; however, at the same time, we also need the right set of people to make it happen.

