Milestone Services Group

Construction Data Migration
Data Quality Assurance & Quality Control

Milestone Services Group
100 Steele Street, Suite 1119
Denver, CO 80206

Contents

Introduction

Simply put complete, accurate and fully functional converted data is the objective of the data migration subproject of an ERP implementation project. While visualizing the objective may be simple the realization is series of intricate steps that require a commitment to quality and the willingness to address the details. The purpose of this white paper is to outline the process by the Data Migration and Implementation teams to produce a level of quality in the data migration that will be a major contributor to the overall success and enthusiastic adoption of the system by the user community.

Maintaining quality in data migration requires that every step, decision and action be filtered through the question; *"Is this likely to improve or degrade the quality of the final data product?"* This question has to be asked because the benefit of any effort that degrades the quality of the data is likely to be accompanies by the time, expense and loss of confidence that almost certainly follows migration of data that in its final state, is of poor quality. Put more succinctly, more data is almost never preferable to high quality data.

As the data migration process involves a long chain of processes and events, quality issues can be injected at any point. For this reason, focusing on the final events in the chain to address quality is a mistake as is the decision to intentionally sacrifice quality now in exchange for a shorter schedule. Low quality decisions are difficult to fully extract down the road and require additional effort to manage. In data migration there is no equivalent to *doing it right the first time*.

Definitions

Before the general discussion of the processes involved in ensuring the final quality of the data, the discussion of a few terms and definitions is necessary. Most of these are recognizable concepts though a review will make the narrative a bit clearer.

First it is important to understand the distinction between Quality Assurance and Quality Control. The goal of quality assurance is to avoid the introduction of errors into the data in the process of migrating it. The application of quality assurance is making sure that every decision, process and program improves the quality of the final data product or at least does not degrade it. Essentially ensuring that the filter discussed in the introduction is constantly applied to the processes and programs as they are designed, built and implemented.

The goal in Quality Control is to reassure ourselves that we were successful in implementing our quality assurance measures. QC involves finding the errors that were inadvertently introduced despite our best efforts. For example, the data validation process is at the center of the quality control process as are a number of other activities.

The following terms are introduced to assist in labeling and defining the various data quality risks.

Completeness – The completeness risk covers the risk of losing one or more business objects during the data conversion. It also includes the risk that additional business objects (such as equipment units) appear in the target system that did not exist in the source system.

Semantics – The semantics risk can appear if a business object has been migrated (e.g. a labor record) but the semantics have changed. The employee hour was originally recorded as a straight time hour but now is an overtime hour in the target system. It can also be that the value remains the same but the unit changes. (e.g. An equipment unit was originally billed for 1 week but appears in the target as 1 day.)

Data Corruption – The corruption risk means that the migrated data does not reflect the data model and design of the target system. This can happen if there are constraints on the application level which are not enforced by the validation scripts or the database. The database and validation scripts do not stop the data from being loaded but the application may malfunction later as a result of the error. This is most common in AP, AR and Subcontracts. Data that is "set in motion" after conversion is the most susceptible to this quality issue.

Orchestration – This quality risk involves the challenges of coordinating all of the data migration tasks and program in the correct order. This quality issue is reduced as much as possible by automation of the process and programs, however, the events leading up to a data load such as mapping data collection and update, clones and cutoff control are not automatable and require planning and attention to detail.

Target Application Parameterization (setup and configuration) - Quality issues in this category originate from the target application. If the target application changes, it can become incompatible with the data transformation programs. For instance, if the Market Sector codes change in the target as a result of further design and refinement by the implementation team, the code and mappings within the conversion programs must change as well to avoid the introduction of errors or improperly mapped data.

Quality Assurance Measures

The key to quality assurance is control of the process. Again the concept is concise and easy to understand but putting it into practice can be challenging given the number of objects and processes that must be controlled. Control in a data conversion project means standardization of methods and the process for maintaining the objects that must be created, after they are developed. These objects include software programs, spreadsheets, mapping tables, parameter values, environments, scope documents and policy decisions.

Scope Documentation

Building on the concept that control of the process is key to maintaining quality, the conversion scope document is one of the first control structures that the DM team (Data Migration team) should assemble at the start of the project. The scope defines the boundaries of the conversion and is used to communicate those boundaries to multiple audiences. The scope document is at the base of the pyramid in terms of controlling the *Completeness* quality issue described above. There are many more programs and objects

that build on the scope document though it is the most important element in the control of this quality factor. The scope is a dynamic document in most implementations. Process and policy decisions affect its content and for this reason it is important to diligently maintain the scope throughout the implementation. Periodic review in a formal setting to verify that it reflects the current position of the company is also necessary. It is the company's responsibility to alert all stakeholders in the project when a change in the conversion scope occurs and to verify that it is reflected in the current version of the scope.

Spreadsheet Maintenance

Throughout the course of the implementation, the DM team will ask the company to populate spreadsheets that contain scrubbed data, data mappings and possibly even master record data (though this is rare) to facilitate the data conversion. Historically these spreadsheet are the source of many of the errors that are introduced in the data conversion process. Many of these errors fall into the *Semantics* category described above. In addition to introducing errors into the converted data, poorly maintained spreadsheets are responsible for schedule slips during mock conversions. Detecting and cleaning up mapping spreadsheets during mock cycles is very time consuming and disrupts the process flow which can introduce additional quality issues. In an effort to eliminate these errors, the following guidelines should be strictly followed throughout the project:

- Spreadsheet formats should be originated by the DM team. If a need arises where the company needs to transmit data of any kind to the DM team, the format of the spreadsheet should be discussed with the DM team prior to assembling the data. This will avoid the need to rework the spreadsheet after the fact.

- Spreadsheet formats should not be modified without discussion with the DM team. This includes the addition, deletion or reordering of columns.

- Valid column values should be verified or approved by DM team prior to populating the spreadsheet. For example, if a column has been left blank because the record did not have a value, it should not be changed to N/A at a later date without the consent of the DM team.

- Rows and columns should be unhidden and expanded before transmitting or turning over spreadsheets to the DM team.

- Updates transmitted to the DM team should be in the form of complete spreadsheets and not incremental changes unless specifically requested by the DM team. This is to say that the entire data set should be transmitted and not additional rows.

Transformation Program Development

The mechanism by which source data is conditioned for import into the target system is the transformation program. The development of these programs is the responsibility of the DM team and these are created using SQL Server stored procedures. The DM team uses a standard methodology for developing and documenting these programs so that they can be maintained and troubleshot efficiently. One of the primary quality assurance tactics the DM team uses is the documentation of data fields that have preliminary or untested

transformation values or rules. These fields are marked in the source code with the work "Verify" so that they can be found in document level searches during the mapping verification and confirmation process. This process involves meeting with the company subject matter experts after the shell program is assembled to verify the values that should be converted to support the company's design. As stated early, it is better to delay the development of some programs until these design issues are settled, however, that is not possible in all cases and place holders must be inserted until the final value can be identified. Note that this placeholder technique can be the source of many quality issues if not controlled in the appropriate manner.

Transformation program maintenance is the modification of the transformation programs after issues are identified in a preliminary review or mock conversion. The DM team should document most program maintenance items with comments in the code that are dated and referenced to the individual or group requesting the change. Some changes are not documented when they do not represent a decision change but instead correct an initial error in the program. This documentation helps both the DM team and the company understand the events that led to the decision to modify a transformation program.

Environment and Parameter Maintenance

The documentation and maintenance of the "Pristine" or "Gold Standard" environment has been a topic of many discussions in implementations. The process of maintaining the parameters and configurations that reflect the company's design decisions is one of the most important quality assurance elements in the project. The quality of this environment and its control is the key to reducing many of the errors that creep into converted data. These errors fall into the *Target Application Parameterization* quality risk previously defined. The setup parameters often define how data will be presented or analyzed in the system in addition to providing the benchmark by which the validation scripts accept or reject converted data. When this environment is not controlled or significant changes are made without alerting the DM team to the change, problems with the quality of the data will most likely occur. For this reason, the company should develop and document an internal process for ensuring that the environment and changes to it are controlled and communicated.

Quality Control Measures

As previously stated quality control is the act of verifying that the conversion of the data happened as intended and serves to initiate actions to correct errors in the data and upstream processes where necessary. *Application Testing* and *Data Validation* are the two primary quality control activities in the data migration. Both of these tasks require careful planning and preparation. The actual tasks, as well as the planning, are the responsibility of the company. Since most organizations do not have experience with these tasks, the DM team should budget significant time to consult and assist the company with the effort.

Data Validation

Following each mock conversion and the final Go Live conversion, the company project team will perform a thorough review and validation of the converted data. This process is essentially the comparison of the converted data to the source data and the documentation of any differences. This is a labor intensive process and one that will consume considerable resources while in process.

The key to the efficient and successful performance of the validation is the development of a thorough validation plan in advance. The data validation plan is an item by item plan for validating the data that is based on the scope. The primary focus of the plan is to define the source and target reports, inquiries and other output that will be used to accomplish the comparison. The validation plan development itself can be challenging when good reporting objects do not exist in the source system. It may even be necessary to develop new reports in the source system to use in the validation. Additionally the data validation plan assigns specific resources to scope items so that the staff knows their area of responsibility in advance of the validation exercise and can prepare for the event.

In some instances electronic comparison methods may be employed. This should be limited to detecting high level variances and the identification of *Completeness* quality issues. The validation effort in large is a manual process and the team should be prepared in advance for this reality. Data validation is rarely viewed as a pleasant task and the temptation to avoid or minimize it must be resisted as it forms the cornerstone of the quality control effort.

Appearance Tests

This testing focuses on the appearance of the data at the Graphical User Interface level. Subject matter experts from the various business areas manually compare data by looking at the GUI data of the source and target. When planning an appearance test, individual subject matter experts identify a concrete sample set of data for testing. They know best which business objects and data items are most crucial. This ensures a high coverage while comparing only a limited number of business objects. If future users (non implementation team members) conduct the tests, they get the opportunity to get acquainted with the target system and often make the most objective testers for this type of test.

Processability Tests

This testing ensures the coordinated and successful interaction of the target system and the newly imported data. Whereas the appearance test examines data at the source and the target user interface (visual comparison) the processability test processes migrated data. This helps with identifying incompatibilities between the migrated data and the parameterization (configuration/setup) of the target system. An example of this is an AP transaction that appears perfectly on the target aging report but crashes the application when the data is selected for inclusion in a check run. These errors are detected only when executing concrete test cases. Also vital for processability tests is to define sample subset of data to ensure a high coverage with a limited amount of test cases.

Integrated Testing

In many cases a business process is not contained by one target system module but spans over several modules and various applications. This requires an end-to-end test or integration test. If one application changes, its interaction with the other applications must be tested. The replacement of a source application with a target application together with a data migration is so significant that the proper functioning of the target system with the migrated data in context of its interlinked applications has to be tested. Originally, data stored in the source application might have been referring to data in other applications. In turn, connected applications were accessing different data sets stored in the source application. The object of Integration testing is to check whether the references between applications still function in both directions when migrated to the target system.

Data Quality Risks

Having discussed the key quality assurance and quality control measures for the data migration, it is advisable to review the data quality risks that were identified in the *Data Migration Risks and Mitigation* document provided by Milestone. These items identify very specifically the conditions that have the highest likelihood of introducing errors into the converted data. Each scenario is followed by risk mitigation strategies and recommended practices that should be considered.

1. **Poorly Defined Transformation Rules** – The embodiment of this risk is data that passes through the validation scripts but does not satisfy the needs of the reporting or system function. This can occur when default values are used to achieve initial milestones but downstream review and mapping does not occur. It can also occur when mapping and review efforts are conducted prior to the subject matter experts or super users understanding the function of the data options. This most often occurs due to schedule pressures and results in the team making incorrect choices. This is a potentially difficult condition for the company implementation team to fully understand until the implementation progresses and actual examples are available.

   Mitigation: In order for the data conversion to be on the leading edge of the project schedule a number of tasks and transformation programs will be built ahead of system design completion. The most important mitigation strategy is to schedule mapping sessions with the subject matter experts following the training and design to confirm the mapping and transformation decisions made by the DM team. These activities should be accounted for in the DM team's scope of work and should be scheduled after commencement and the initial design workshops have been completed in each area. It may be possible to schedule those meetings sooner but a more completely defined projects schedule will be needed to facilitate that.

2. **Overly Complex Transformation Schemes** – Of course the term "overly complex" is a subjective phrase and potentially hard to define exactly what is too complex, however the concept is not difficult to understand. In some instances implementation teams become overly aggressive in their attempt to correct or improve data which in its native state, is poor or incomplete at best. This leads to transformation programs that may produce data that is not able to be validated

because it deviates from its natural state to the extent that it is unrecognizable and therefore its validity is undeterminable. Additionally using complex transformations to create new reporting structures (divisions, departments, and even companies) that did not exist in the legacy system often lead to serious data integrity and quality problems and should be evaluated on a case by case basis.

Mitigation: The best mitigation strategy is to keep the purpose of the conversion in perspective. Converted data is not always "forever" data. In many cases is usefulness is high for some period of time following the conversion but that usefulness evaporates as time goes on. It is inadvisable to create transformation schemes where the complexity creates problems that may outlive the usefulness of the data. The DM team should provide guidance in this area to help the company determine what level of complexity is appropriate for the need.

3. **Conversion Validation Script Inadequacies** - It is possible for the validation scripts to apply inadequate validation or in some cases incorrect validation rules to the transformed data. The target application is typically complex and the validation rules may not always be consistent with the application validation. The validation procedures often operate independently of the business application layer of the target system and have the potential to allow invalid data to pass through. This is not commonly a significant issue but it does happen.

   Mitigation: The strategy here is for the validation team to do the best job possible of reviewing and testing with the converted data to make sure that it functions with the target application as it should. These problems are most often detected in testing cycles and are easily corrected if properly documented and communicated.

4. **Design Changes** – Alterations in the design of the software or modifications to the function of the software have the potential to make once valid data invalid in the context of the enhanced or modified functionality. This most often occurs when a process team makes a significant design decision after the mapping sessions and fails to communicate that to the DM team. This has the potential to occur throughout the implementation but has a higher likelihood of occurring toward the end of the implementation when rigorous testing and close scrutiny is applied to the design. As an example, the worst-case Milestone has encountered was the last-minute addition of several companies to the conversion. This event occurred a couple of weeks prior to Go Live with the company failing to communicate the decision and the result was that the conversion programs had to be modified during Go Live to accommodate the change. Needles to say this resulted in a stressful situation for all involved.

   Mitigation: Again, communication is the key to controlling this risk. The implementation team should assume that significant changes to the established design will affect the data conversion until confirmed otherwise. It does not take an unreasonable amount of time to incorporate this check into the standard operating procedure and it will improve the data quality many fold.

5. **Inadequate Validation** – The highest risk factor associated with poor data quality is inadequate validation of the converted data. If inadequacies or errors are detected they are almost 100% correctible. Undetected data issues will invariably show themselves at some point down the time line after Go Live if they are significant at which time the correction process may be difficult and inconvenient for the users. This situation occurs when the proper importance on the data validation effort is not emphasized or the validation team makes the assumption that they are not responsible for errors if they do occur. As previously stated, the ultimate responsibility for data validation lies with company. The DM team's responsibility is to carefully execute the conversion plan and process and educate the company team on the best practices for validating the data. Quality is important to the DM team though validating the data at the level that is necessary is impossible for anyone other than the data owners.

   Mitigation: Communication of the responsibility to the validation team and potential consequences of failing to devote the time and effort to a comprehensive evaluation is the first step in controlling this risk. Additionally, allowing for adequate time and resources to develop the validation plan and train on its use is essential. A well trained team armed with a solid validation plan will make all of the difference in the quality of the data conversion effort. The DM team should cooperate and work with the company at the highest level to ensure that the issues are corrected and that the data supports the needs of the project.

Summary

Data quality issues encompass a broad range of risks that range from the absence of data from the scope that is ultimately needed to successfully transition to the new system to the incorrect transformation of data resulting in incorrect values. The adage that *prevention is better than cure* should be the motto for the data conversion project. Sound practices, solid control of the processes, mapping objects and transformation programs (Quality Assurance) will make the quality control aspect of the project much less labor intensive.

While no data conversion is perfect and migrated data tends to lack the full attributes of system generated data, quality issues can be overcome if discovered during the implementation and corrected at the source of the problem prior to the go live conversion.