TITAN

CONSULTING

SAP DATA MIGRATION

SAP DATA MIGRATION OVERVIEW

OVERVIEW

A key part to any SAP project is the Data Migration process. Considering the importance, much of the work with Data Migration has to wait until later in the project. This results in time constraints, and many times results in short cuts being taken. Often, the data migration can contribute heavily to how successful the cutover is considered by the users.

FUNCTIONAL REQUIREMENTS DEFINITION

Unfortunately, data requirements are typically one of the last defined objects in an implementation. It is impossible to start defining these requirements until the functional blueprint and design phases are complete. Then the requirements will continue to change based upon the results of the various testing cycles. It is very much a “living” process that evolves through the course of the project. The term “code freeze” is often desired and rarely accomplished in data migration.

As an example, if a critical defect is raised in the final testing cycle that impacts the data, the migration team must adjust to accommodate the need. There is no choice. In addition, the data in the legacy system changes daily. Many scenarios will show up that were not anticipated. For instance, a brand new supply chain agreement with a vendor for consigned inventory may arise two weeks before the SAP system goes live, thus resulting in a change to the mapping to support consignment.

There are typically three documents to support the requirements behind the data migration activities that are created per conversion object. Once again, these cannot be created until the blueprint and design phases are complete.

The first is a functional specification that defines the functional requirements of the upload program on the SAP side. This usually contains screen shots and information about how the data would be captured in SAP if entered manually by a user, including behavioral nuances of the transaction for different types of data in the same transaction. As an example, different order types may spawn different screen sequences and data elements for the same base transaction.

The second is a technical specification that defines the specific tables and fields that are to be updated and any logical flow details. These requirements are stated in terms suitable for a developer to begin the actual development of the upload program.

Once the upload program is written, the file layout that is required by the upload program is then fed into the third document that is called a mapping document. For each field in the mapping document, a source field, transformation logic, or constant values are identified. Data mapping sessions are needed to complete the mapping documents and require resources from both the SAP functional side and the legacy side. The Data Migration team uses the mapping documents to create programs that will transform the source data into files that the upload programs can process.

PROGRAMMING REQUIREMENTS, DATA TRANSFORMATION AND LOADING

Migrating data into SAP will require programming and/or data tools on both the SAP Import and legacy extract sides. On the SAP side, there is a powerful platform called LSMW which will accommodate loading requirements of most of the major objects in SAP. LSMW is a very flexible tool and has the ability to call a wide variety of standard SAP upload programs and functions. It is basically a front end wrapper program that allows the developer access to a plethora of SAP provided objects including ABAP programs (BDC and Direct Input), BAPIs, and IDOCs, as well as user defined transaction recordings, and associate it with an input file structure.

One major consideration when deciding to use the LSMW platform is to decide where the transformation logic will reside. LSMW does allow for complex logic through ABAP code to be included in the conversion process. The end result is the LSMW program itself will not only upload the data into SAP, but it will also contain some or all of the mapping rules and transformation logic. In the end, you will have a program that is for a very specific purpose and is not reusable in its current state for future deployments. In addition, if there are complex mapping rules in the LSMW, a detailed review and modification of each rule is required before it can be used for a future deployment. If anything is missed, a rule could be in place where it is not needed or wanted, thus passing unintentional erroneous data into SAP.

The other option is to create basic LSMWs that simply pass through the data that is provided in the input file with little to no transformation logic built in. This allows for a library of LMSW’s to be created up front and reused again and again, with little to no modification. It also puts the transformation logic work squarely on the shoulders of the resources that are extracting the legacy data and creating the input files. These are typically legacy data experts who are in a much better position to implement complex logic than SAP developers who may not be familiar with the legacy systems.

In addition to utilizing LSMW, there are several other viable solutions for uploading data into SAP. These include options such as custom and standard ABAP upload programs, CATT (Computer Aided Test Tool) scripts and Batch Input Recordings. There are also 3rd party tools that provide Excel based data collection that is then pushed into SAP. In most cases, the SAP technical developer will choose the method and create an upload program that accommodates the requirements in the technical specification.

On the legacy data extraction side, there are also several options ranging from low cost, highly flexible in house solutions to full ETL platforms. A typical in-house solution would involve having a SQL database created that will hold the raw data from the legacy systems into a "staging area". Then a second database is created that contains target tables for the transformed data that closely resembles the target tables in SAP. This database also holds any cross reference or lookup tables that are required for data transformation.

It is preferable to have a third database created that will hold extracted data from SAP. This is used for a wide variety of purposes including looking up SAP internal keys (material numbers, customer numbers, etc…) and allowing the ability to create delta runs. When you can see the data that is in SAP as part of the overall data migration landscape, the benefits are great from not only a data migration point of view, but also from a data validation point of view. It is not recommended to go straight to the database tables behind the SAP application. Therefore, it is a requirement to have the data pulled or pushed from SAP itself. This can be done in numerous ways. A straightforward way is to have an ABAP program written that allows the user to select the tables and fields that they require along with selection options. The program will then periodically extract the data from SAP into files that reside in directories on the application servers. These files can then be picked up by a SQL job and loaded into the data migration area. The advantage to this method is that only one ABAP program needs to be written and more tables and fields can be added as needed, on demand, by the Data Migration team.

With the legacy data and the SAP data in hand, the last step is writing the scripts that produce the output files that are ready to load into SAP. In this scenario, everything was built in-house to manage the data migration effort. One last consideration when using this method is to implement source code version control. Subversion (SVN) is a very good, open source solution that is available free for download at http://subversion.apache.org/. There are other source code control options on the market as well. This approach provides a robust tool set that will allow for full documentation that is attached to each script. This is valuable as the requirements change and also provides the ability to go back to previous versions of the code in the event that the requested changes did not yield the desired results during testing. It also ensures that the code is consolidated on a server and not on individual workstations, resulting in a low probability of lost work and reducing the impact when human capital changes occur.

Alternately, some companies will choose to go the route of using an ETL package. This provides all of the functionality of the above mentioned approach along with several other features. There is still quite a bit of set up for the environment and the costs can be high, requiring full system landscapes that mirror the SAP development, test and production instances of SAP and specialized consultants that know the ETL tool. They are procedural in nature and offer a high level of control as well as other numerous features that are nice in assisting the building of quality data sets. In addition, once the ETL package is purchased, it can also be utilized for many other IT data initiatives that are not related to SAP data migration.

LEGACY DATA CLEANSING

Data cleansing is a critical part to any successful implementation. A perfectly designed SAP system and perfectly written conversions will not yield a favorable result when loaded into SAP if the data in the source system is anything less than perfect as well. To assume that the data in the legacy systems should to be converted "as is" is a bad assumption. It is very typical to have errors, creative users who have found "workarounds" and many other scenarios that would violate the logic that has been put in place, resulting in unfavorable results.

It is possible to put ten people in a room and ask each one to define what an open order is, or how serialized material is reflected in the legacy system, and get ten different answers or find that there are exceptions to the rule. The data mapping sessions highlight many of these issues. Then the cleansing tasks start coming to light.

Another example is there may be ten duplicate customers on the legacy side that have been created in error over the years. You would want to reflect them as a single customer in SAP; therefore, they must be grouped. There may be open purchase, manufacturing or sales orders that are years old or in the future and not closed. There are cleansing tasks that can be defined for just about every data object in SAP. These will get addressed one way or the other. It is a huge advantage to address them as part of data cleansing as opposed to fixing it on the other side in SAP after go live.

That being said, not all of the cleansing tasks are a result of bad legacy data. There are some fields in SAP that simply don't have a corresponding legacy value. In these cases the values must be collected by other means. This is also part of data staging and cleansing.

SEQUENCING AND CUTOVER

Data migration requires a specific sequence of load activities which are usually documented as a subset of the Cutover Plan. Some are common sense, such as you can't load a purchase order until the materials and vendors are in place. Typically the master data comes first, then the transactional. However, when creating a migration plan, there are numerous other things that come into play, such as validation. Validating and reconciling inventory dollars can be a time consuming task; therefore, you may choose to load the master data objects materials, material costs and warehouse bins and the transactional data of batches and inventory quantities before you load the master data for vendors or customers, for example. If customer supplied inventory is required, then customers must also be loaded before inventory as well.

You may choose to load the majority of the master data several weeks ahead of the actual cutover weekend. This reduces the amount of work required at cutover, but also requires dual entry of new objects in both the legacy and the SAP system and delta processes to ensure that all of the data has been maintained directly.

There are also many issues can arise specifically because of how SAP works or is configured. An example is the Purchasing Information Record (PIR) in SAP. This is a piece of master data that contains information that, in some cases, can influence the Standard Cost calculation as part of the SAP Costing Run. If the PIR's are loaded before the costing run (which sets the standard cost in SAP), the PIR price (based upon configuration) will be set to standard instead of moving over the standard cost from the legacy system. If this happens, there could be a significant dollar variance when converting inventory that could have been avoided by converting the PIR's after the costing run.

In the end, the specific project requirements and configuration will drive some of the sequencing, while prerequisites will drive others. Some tasks can be run concurrently, while others will need to be run in succession. A knowledgeable resource that has the background to take all of these challenges into consideration is needed to create a well-sequenced and optimized load plan that will ensure that all activities can be completed with the desired results during the cutover window.

CONCLUSIONS

Successful migration of data into SAP is a sizeable task that should not be underestimated and should be planned from the beginning of the project. There are many challenges that are not covered in this document that most companies will come across during their implementations. Allow for these challenges to be visible to the program management team. Implement an issue resolution process and tool, and, most of all, allow some budget for an experienced data migration consultant. Many of the challenges that a company will face during their implementation have been faced before. There is no reason to reinvent the wheel. You will find that the investment is well worth it.

DEFINITIONS

ABAP (Advanced Business Application Programming) – This is the proprietary programing language of SAP.

BAPI (Business Application Programming Interface) – This is another structure that is used as a vehicle in SAP for transactions in a synchronous manner.

BDC (Batch Data Communication) – This is an ABAP program that uses call transactions like a recording. They are flexible and logic can be built in, but the performance is typically inferior to calling a BAPI or a Direct Input program.

ETL (Extract, Transform and Load) – ETL platforms are programs that have been developed by 3rd parties and support 3 distinct functions that are required for data migration. The extract function pulls data from the source system database. The transformation function is where the complex mapping logic resides. The load function pushes the data, once transformed into the desired state, into a target system's database or into files that can be uploaded into a target system.

IDOC (Intermediate Document) – This is a structure used as a vehicle in SAP for asynchronous transaction processing, meaning that each individual transaction is processed and a reply is received before proceeding to the next transaction. IDOC's are a common vehicle for processing EDI messages in SAP.

LSMW (Legacy System Migration Workbench) – This is a tool provided by SAP for purposes of migrating data into SAP.

Recordings – In SAP, a recording is like a macro. The user processes a transaction. SAP records all of the keystrokes for the transaction. At that point, the user can create a flat file with multiple transactions that are fed into the template recording for mass processing.

unleash the power of together®

www.titanconsulting.net

GLOBAL HEADQUARTERS

3411 Preston Rd., Ste. C13-PMB 205 / Frisco, Texas 75034 / Main: 972.377.3500 / Fax: 972.692.7436