YOUR ULTIMATE CHECKLIST

DATA MIGRATION

Software SOLVED®

IDENTIFY WHY YOU ARE DOING THIS

This might seem like an obvious first point. Clarity on why the data is being migrated, what data is being migrated and what benefits you're looking to gain. This will help everyone on the project to focus on goals and use the most appropriate implementation technology and techniques.

UNDERSTAND THE EXISTING DATA

We have seen this happen several times. A data migration project starts and those involved don't have a strong understanding of the existing data. This results in timelines dragging out as new challenges are found that break existing assumptions. Before any implementation design can be done, a good understanding of the existing data is essential.

PREPARING ANY DATA CLEANING, FILTERING, AND ANONYMISATION

Some projects will take the source data as is, because it's known to be of good quality. For some data sources, cleaning and filtering are required so that the target location or system receives high-quality data, and we don't end up with a rubbish-in-rubbish-out situation.

There are also going to be scenarios where the source data should be anonymised as part of the migration. This should be considered at this stage.

SECURITY

Security considerations are very much a part of data migration projects. How will you get access to the source data? Is the data secure as it's being migrated? Will the same levels of security apply to the new location, or have you accidentally exposed data from the source system to a wider audience than planned?

TIMING AND FREQUENCY

When you're performing a regular, recurring migration, you need to be clear on how frequently data will be migrated and when the migrations are going to happen. You might only have a certain system downtime window in which to complete all the data migration work.

DATA UPDATE STRATEGY

If you're running recurring data migrations, how are you going to update and/or remove existing data in the destination location? While it's not the most efficient use of resources, in certain circumstances, performing a full wipe of the destination and replacement is the simplest logic to implement. It doesn't scale at all well as your data volumes go up, so planning an appropriate incremental update strategy is needed most of the time.

YOUR ULTIMATE DATA MIGRATIONS CHECKLIST

DATA MIGRATION

Software SOLVED

RESOURCE REQUIREMENTS

Key areas to think about are the strain on the source data source. This often feeds into the timing of the migration as the extra load of the data migration slows down access to the source data for existing users. Likewise, the added strain to the destination data source of loading in all the new data impacts performance for users of the destination location too. There is also the impact on the network between the two systems that has to be taken into account.

PROGRESS AND ERROR REPORTING

How are you going to track the progress of data migration?
How are you going to handle error reporting?

FALLBACK TO 'KNOWN GOOD' IN CASE OF FAILURE

Even in the 'incremental' data migration approach, there's going to be times where the migration might complete without any obvious failure, but on subsequent checking it's clear that something has gone wrong. In these circumstances the destination data source should be reverted back to the state it was in, prior to the migration running.

VALIDATION CHECKS AND VERIFICATION

Once a data migration has been completed, how do you know that it has been completed successfully? By virtue of no errors occurring during the migration.
There are usually several checks that you can perform, such as comparing the record counts of the source and destination data sources. You could also review row counts in the destination, before and after the data migration. If these row counts are always expected to go up, then a validation rule can be created to check that this is the case. If not, a rule can still be created that ensures the growth or shrinkage of record counts doesn't go beyond a specified percentage.

UPDATE DOCUMENTATION

It's not the most exciting part of a project, but don't forget to update the system documentation about the migration process. If this was a one-off data migration, update the existing documentation so that everyone knows where the new data location is.

DECOMMISSIONING OLD SYSTEMS OR OLD MIGRATION PROCESSES

Lastly, don't forget the decommissioning of old systems or migration processes. Once a new environment or migration process is stable, it's a good time to turn off and uninstall the old system environment.