migrating complex data.
faster, cheaper, better.

Thoughts on complex data migration

A paper for data migration professionals

Abstract: *Data migration is a complex process, requiring a robust methodology and good supporting tools or software to deliver a high-quality result as fast and cost efficient as possibly.*

We hope that these thoughts will help you understand the challenges and risk inherent in any data migration project, but also understand that there are better and worse ways of meeting these challenges and managing the risks. A data migration project can be complicated and time-consuming, but if you do the necessary preparation work and fully think through the process, you should be able to avoid some common data migration errors.

There have been a lot of advances in data migration, and you really need to look for tools that help automate the migration process.

Our common challenge

Data migrations have taken place as long as new systems or processes have replaced older.

Like when Gutenberg invented the printing press and all the old hand-written books had to be 'migrated' to the new technology or like when medical records where moved from being stored on paper to digital media.

Examples are endless and yet the industry doesn't have a common good practice. In many ways an overlooked and immature corner of the otherwise advanced IT industry.

Let's start with what we do know about what's good and not so good practice.

What's good practice?

Many best practices of data migration focus on its iterative nature. Development is an iterative cycle profiling legacy data, modelling target system, developing transformations, and creating test cases.

Best practices also involve engaging end users in the development process, coordinating the migration's rollout with a new system implementation programme, 'synchronising' old and new systems, handing off the new platform to its new administrators, and retiring the legacy platform.

Some of these tasks are business issues and will require collaboration between IT's migration team and the business users affected by the migrated systems.

Combining all of these issues is what makes data migration a complex and demanding activity that's best accomplished using a feature-rich data integration tool that supports iterative methodologies.

hopp tech ltd
86-90 Paul Street
London EC2A 4NE
United Kingdom

t +44 (0)203 322 1718
e info@hopp.tech
w www.hopp.tech

Registered at the above address
Company number: 10421313

What's bad practice?

All too often data migration projects have no methodology, no technology, no clear deliverables, limited budget and a hard deadline.

Some people say there is no need for subject matter experts as data migration is perceived to be a straightforward exercise moving data to a known system.

Very often the data migration is viewed purely as a data exercise – something that takes place in the IT department, and often tagged on to the end of the much higher profile new system implementation.

The fact is that the new system may look nice and exciting, but if you're not assigning the right people to the job – that is all it will ever be!

Background

When considering the implementation of a new system everyone involved from management to users are usually eager to understand how best to manage the transfer of their legacy data to the new system - whether it's done by a vendor, consultants or its own IT organisation.

This paper highlights the key areas concerning methodology and discusses several questions to be addressed in any data migration project.

It's our aim to give you some sound generic answers that we consider to be current industry practice and where our approach differs show you how and why we believe our approach provides additional benefits in our common challenge in delivering data migrations faster, cheaper and better.

LEARN FROM OTHERS

... AND CONTINUE TO INNOVATE!

Our take on the challenges are an attempt to improve the existing practice - which we feel could be much improved.

By integrating our approach into a software solution (migFx - Data Migration Framework) we argue that you have a better approach than the current standard practice across the industry.

The Challenge

Data migration – typically defined as the process of transferring data from a legacy system to new system.

While it may sound a rather simple exercise it very often turns out more difficult in practice (e.g. most data migrations projects partly fail on cost, time and/or quality).

The more complex the data and the logic behind the data, the more challenging the data migration processes become. If the actual migration is not prepared well enough, for instance not mapped, validated and tested, the data migration might be quite destructive and not what was expected and end up costing more and risking the whole system implementation.

Following any step by step guideline will not guarantee success, but it might help highlight common pitfalls.

While every project is different, we believe is that a common approach to the process of data migration can be adopted, and it's worth thinking about best practice so you can learn from the mistakes of others.

We all try to learn so that next time we can deliver data migration better, cheaper and faster.

This white paper outline lessons learnt and what project approach we recommend to ensure a successful data migration project.

Think twice

Organisations' needs for system support are changing continuously. Successful data migrations help organisations get ahead - with systems prepared for new initiatives to benefit users and customers.

Data in and of itself has little value. What makes data matter is how it's used! Organisations use their data to make decisions, deliver their products

and services, support customers, remember past events etc.

Before you approach data migration, think about how the organisation uses its data and for what. Focus on the business processes the data is linked to as well as the content of the data.

It's crucial to a successful data migration to know how the data is to be structured and used in the new system – and to ensure no knowledge is accidentally lost in the migration. Think about what data is not migrated as well as what is. We call the unique structure and use of data the 'business logic' behind the data.

THINK BEFORE YOU START!

Not knowing or not caring about the business logic could jeopardise the success of your data migration project in a way that could catastrophically impact the user's ability do their work post migration.

Just remember data migration is as much about people, customers and processes as it is about data and systems.

As we said before, data migration is often a very complex process which requires good cooperation among people across the organisation both business and technical. So remember to communicate with everyone involved or affected by changes introduced by the project.

Some useful definitions

There are many situations where data must be moved for different reasons, such as:

- *Data Consolidation or Integration – this is the process of combining multiple data sources into one or fewer as seen in data warehouse solutions and will usually involve frequent updates.*
- *Data Migration – this occurs when data is moved from one or more systems/data models to another as typically seen in one off system changes or a system upgrade.*

The main distinction is the frequency and reason to move not only data, but also the extension of business rules and functionality to another system/data model.

- *Complexity - by it we mean the diversity of source and target data models and the associated business rules.*

Much of the time spent in data migration projects concerns the specification and development of mappings and transformations between source and target systems. The more divergent they are (in terms of number of data models or business rules), the more time analysis, specification, development, testing, training etc. – (i.e. the project) will take.

Preparing a strategy, following best practices, and avoiding common problems increases the probability of success in any data migration.

We believe that the lessons and good practices listed in this paper will help you achieve this goal.

Bigger picture / Moving target

The reality is that it doesn't matter if the target is firmed up yet start as soon as possible, it will always take longer than you think. At the outset don't sit around waiting for a complete technical specification for the target schema, functions and business processes. However, it will matter later in the project if the implementation team can't keep their lid on the design changes.

Scope the project

While not attempting to rewrite the book on project management here – a data migration project is a project like any other and it is crucial that its well planned, managed and executed - therefore a few words on project management before we move on.

To start the project, your scoping should identify potential issues and risks that may occur later. This enables the migration team to mitigate known risks.

The aim of any scoping is to thoroughly define and allow stakeholders to review the project before it starts.

We believe that such a review should cover the project's structure, approach, requirements and its technical aspects.

Focusing solely on the core methodology issues the project review should highlight the following areas:

- *Which alternative methodologies have been considered?*
- *Are the chosen migration methodology and tools fit for purpose?*
- *Does the migration tool fit with the skills of the people assigned to the project?*
- *Are there enough team members and do they have the right skills?*
- *Is there a data quality plan?*
- *What is the overall project timeline and activities based on?*
- *Are the timelines realistic given readiness, methodology, tools and resources?*
- *What are the technical features of the proposed data migration tool?*
- *How is the success of the migration to be judged, who will sign off the data once it has been migrated?*

The list above is a supplement to the typical project challenges in any large complex project, managing stakeholders, communication, and change. A data migration project adds to these several critical issues involving methodology and tools with which most organisations have little experience.

Have a plan

Although data migration is not always the main activity in the overall project plan (i.e. new system implementation), the main migration activities need to be separately mapped and project managed to ensure the needed focus.

Project initiation and training is important to ensure that everyone knows what is expected of them and that they can do the job, especially in data migration projects which will be new to many.

As in any well managed project map the sequence of tasks, deliverables and dependencies and assign people to each activity. Check that you have the right resources to complete those tasks.

Most projects struggle with matching people, skills and tasks so understanding what resources you need to accomplish the job will help you be fully prepared.

PLAN FOR THE UNKNOWN

Tasks, people and skills required are closely linked to the choice of methodology. We challenge you to consider how you make a credible plan for activities you are not familiar with or never tried before?

How realistic is your plan when you just don't know?

What usually happens is that you have an overall milestone plan and ask your teams to meet the deadlines?

Most project plans will have some ideas on dates or timelines indicating when the teams require to be where or when certain activities should be completed.

What this will not show you is the overall workflow that will get you to those points. Although the iterative nature of data migration projects makes it difficult to plan in detail the key milestones and the dependencies within the overall programme plan need to be defined before project start so that there is no confusion as you move into the project phase.

It will also help you identify gaps in your resourcing need where the necessary skills or budgets are lacking.

How good is your plan?

If you have followed common practice you should have a reasonable level of detail for your project plan.

Having recognised that planning without perfect knowledge contains the risk that things might not be as expected. Several areas can throw surprises at you in data migration projects.

- *Data quality not as expected*
- *Activities takes longer than expected*
- *New requirements are generated by the system implementation part of the project*
- *Required data is not available in the legacy systems*
- *Changes in the overall programme plan can change deadlines*
- *People can be unavailable/unhelpful or reassigned elsewhere because of other priorities*

Not criticising, but it's always difficult to plan the unknown. Therefore, time needs to be spend on understanding these unknowns.

The best way is to use a methodology that allow you to try, test and learn fast using an iterative agile and incremental process - ideally supported by an easy to use tool.

You just can't create accurate timelines before you have looked at the data and complexity of the business logic being migrated

---

For example, simply creating an arbitrary 4-week window for "data quality activities" is meaningless if the data is found to be truly abysmal.

It is vital that you understand the dependencies in data and business logic, before you start mapping, and you can't do that until the analysis and discovery phase has completed.

Don't just rely on someone else's previous data migration project plan, your plan will be dictated by the conditions found on in your specific project and the wider programme that your project sits within.

Choose a methodology

A well-suited methodology is an important part of a successful data migration. As there is no common industry approach most organisations have their own methodology or none at all.

The traditional approach includes manually coding your migration using a combination of scripts, SQL and home-grown routines hopefully based on detailed textual mapping specifications made by business analysts. It may appear common sense and cost-effective, but there are a host of reasons why it may be time to change if you are using such approach.

More recently teams have tried to adopt an Agile approach to the development of custom code. While this is more suited to the iterative nature of data migration projects it can lead to an unstructured codebase that is difficult to amend in the later iterations.

In either case when you write your own code from the ground up, it might work better than things prepared by others. It would surely be yours and highly tailored, but can you guarantee that it will be free from bugs and that everyone else will be able to use it? Not really, right!

The best applications developed by any organisation take thousands of hours to develop and test. Can you afford that with your own code? Probably not. Therefore, if possible choose tools already there and enjoy using all the work someone else has done for you.

Using an appropriate set of tools can help you become more productive – saving lots of effort, money and time while delivering better quality.

Tracing data lineage, loss of key resources, version control management, limited data quality features, speed, complexity - there are many good reasons why it's time to explore tools and technology that could accelerate and simplify your next migration.

We find it critical that the project approach (next section) and the chosen methodology work together and not against each other. Our methodology consists of a set of proven software tools and a complimentary project process that leverage both tools and process to the full extent.

Supported by proper project planning our methodology will provide you with the tool you need.

Project approach

Are you still thinking about applying a waterfall approach for your data migration projects? If yes, then again, it's time to change.

Data migrations do not suit a waterfall approach, yet the majority of data migration projects still follow a classic waterfall design.

Agile, iterative project planning and execution with highly focused delivery drops are far more effective, but you must ensure that your overall plan is flexible enough to cope with the likely changes that will occur and allow you to learn and improve as you go.

So, design your data migration projects around results-focused iterations moves your project in increments - that works!

Think agile and build rapid discovery elements into each iteration and plan your project based on what you discover, not what you thought you knew.

Don't make the mistake of planning your entire project at the outset, you simply don't know enough to start with.

Convince your stakeholders that an agile, transparent approach is best placed for managing risk and delivering a data migration that the business will benefit from.

So, if you are going to use a toolset to implement your project it must support an Agile approach to your project

Data Quality - Profiling

The big question here is – do we have the data needed to make the new system works as intended? Are our data complete and of good quality?

Expect data migration to expose data quality issues. Some of these are problems requiring a fix, while others are opportunities for enriching data. Either way, raise the bar to improve data and metadata as you migrate them.

If not, then your data migration project will have a significant challenge to address before you get very far.

How to best approach this challenge?

The business should know where all the current data are used and what data is needed to make the new system/process work as intended. They should also know about the shortcomings of the existing data, where are they feeling pain now and how do they hope the new system will alleviate that. So, early identification of all possible data sources and requirements is vital to ensure accurate and successful data that will enable the new system to deliver the benefits identified.

At this stage, we discover and make all the data available with their known relationships to one another. We look at the data to understand what challenges they may hold. Working backwards from the new system/process is the best way to identify the critical requirements – never start with 'this is our data, where should they go'.

Once we have collected all the data needed we need to understand how it should be transformed to meet the requirements of the new system ready to migrate. In an ideal world, you would migrate 'everything' however the reality is that there will be data which may not add value or may be covered by a different source.

Data quality rules should be used to measure the quality of the data and to help fix or mitigate any data quality issues. Every data quality issue should have a data quality rule decision made as to how to obtain an acceptable data quality in the target system.

Once data quality rules are identified they should be built into the migration process so that they are measured each time you iterate through a set of data.

Options available for fixing data quality issues:

Ignore

Not significant for the migration and can be fixed later in target system.

Correct during the migration

Add a rule to transform into a more manageable or final solution. This has the disadvantage of making the transformation more complex and potentially the reconciliation more difficult.

Correct in source system

This is usually the best solution especially if it requires user input.

Correct in target system

The issues can be postponed and maybe fixed in the target system, provided it doesn't impair the use of the new system.

Use tools don't try to manual code it

Hand coding still appeals to some, despite a lack of productivity because developers can't wean themselves off it, consultancies use it as to rack up costly hours, and short-sighted managers won't spend in the near term to get the long-term cost reductions of tool productivity.

For larger data migration projects, there is no question - tool based specification, development and deployment, is far more productive than hand-coded solutions.

Multiple solutions, tools and technologies are available for data migrations, and more than one of these tools may be useful in a single project. Every data migration will have a unique combination of pre-existing systems, plus requirements for the new system.

Organisations should select tools based on these factors, while considering how often and deeply they will perform data migration and similar projects in the future.

Tools for data migration can be grouped as:

1. *Simple Database replication tools*

   Easy and accessible but miss important requirements for use in complex data migrations. They are often limited to moving data one way without transformation like instances of the same database brand.

2. *Extract, transform, and load (ETL)*

   Tools are designed for data integration, but are often used for data migration. Organisations might choose ETL tools for their ability to handle the extreme requirements like large datasets, data trans-formations, deep data profiling, and data integration capabilities.

We have developed a toolset, MigFx, which is specifically targeted at transformation for Data Migration. This means that its functionality includes features that support the Agile project methodology we have been discussing, while providing a structure that supports the management and control of the project.

Getting on with the job

This is where we start working by analysing requirements and quantifying any data gaps. There may be issues in the legacy systems some of which will already be known to the end users.

Other issues will only be discovered when the source data is examined for migration or even later when source date are inserted into the new system.

It is important that these issues are 'flushed out' as early as possible so that appropriate action can be taken.

Detailed mapping

The mapping phase is often the most time consuming, but also the most important in any data migration project. It's here you define how the individual data elements should be linked from the source to target system. All objects should be mapped, down to attribute level.

Traditionally this would be done by a business analyst and documented in long written specifications. The document would then have to be passed to a developer for implementation (coding) in a ETL type tool or manually coded all together, then tested and looped back for clarification etc.

Alternatively, the developer has to talk directly to the business SMEs and prototype the code over multiple iterations, in this case little or no documentation is generated. This process is fraught with the possibility of misunderstanding and ambiguities which may not be discovered until late in the process or worse after go live.

Even in straightforward data integration projects, data mapping is never as simple as just 'connecting the dots'. With data migration, relatively little data is merely mapped and copied. Most mappings include data transformations, because source and target data models are always different.

Transformations of legacy data tend to be complex, because some source fields contain business logic that must be identified and mapped separately. Furthermore, legacy data could have missing values that have to be filled in from other sources. And then there are the usual transformations for standardization and quality, plus rules for conflict resolution, exceptions, and branching.

As a result, data mappings are always complex, consisting of many components that must be designed and tested individually before being strung together in a data flow. Given the complexity and the iterative design method, users should avoid hand coding and do this work in the GUI of a dedicated data integration tool.

WE MAP DIFFERENTLY

In our migFx data migration framework the mapping and coding is one and the same thing. As a business analyst, you map using our tool called Studio and the code is then auto generated by our Engine - done. No loops here between the business analyst who is doing the mapping and the developer doing the code. There need be nothing to manually code.

In projects not using migFx the team managing the migration execution may not be the team responsible for coding the migration logic which brings us on to the importance of managing execution and flow.

Execution and flow

It is essential that the mapping and rules that are used to link the source and target systems are accurately published, coded and the correct flows are established. This will allow the execution team to analyse the root-cause of any subsequent issues discovered. As noted multiple teams are now involved in the process from the business analyst to the developer to the execution team, and traditionally project management, communication and common reference are crucial to progress and quality.

WE EXECUTE DIFFERENTLY

Using migFx the components Engine and Director lets you (the user) generate the code and execute it at a very granular level by yourself - making testing, error and event handling fast and easy allowing a rapid interactive process to be followed. We cut out the need for multiple handovers between teams of analysist, developers and execution with all the attendant delays and opportunities for misunderstandings.

Focus on refining data migration mapping and validations in an iterative process and you will continue improving these until the final migration run. Ideally you should be able to view and test the data in the target system as part of the final iterations.

The traditional waterfall project models don't easily allow for multiple iterations which makes learning and continuous improvement difficult in data migrations projects that don't follow an iterative approach.

Testing and tracking

Final testing is almost as important as data. You are not finished with success until you test everything in the target system.

Moreover, it is always better to assign some testing tasks to actual users and other people involved from the business side. This way, you ensure the most accurate possible control and proper evaluation.

WE TEST DIFFERENTLY

The Tracker component of the migFx toolset allows the business SMEs to review and investigate issue identified with the data and review the source and target data at each stage to diagnose the problems. The results of each stage of the transformation process are available in a simple to use browser-based interface that anyone can access.

It's quite a common habit to wait until the job is done to validate its results. In some cases it is the only option. However, the sooner errors and mistakes are identified, the easier it is to correct them or come up with a hot-fix. Thereupon, one of the best practices in data migration is to keep an eye on the migration during the process itself. This way, it's possible to save a lot of time and - if it says more - really a lot of money.

In an ideal world, a data migration project will follow software engineering best practices, like isolating separate environments for development, test, and deployment. With data migration, however, the deployment environment is usually the target application. And development is inherently entwined with data profiling and iterative testing of mappings and transformations, so it often makes sense to consolidate data migration development and testing into one environment.

Reporting progress

You need to ensure that enough reporting is in place so that everyone involved in the project can see the progress and issues that affects them or require them to respond.

It could be, a project manager who needs to see daily reports or a IT operator who wishes to see current runtime issues or a business leader interested in overall progress on a weekly basis etc.

Interface specification

In our terminology and solution, the interfaces needed are typically the interfaces needed to extract the data from your legacy systems and the interface required to load the data into the target systems.

After the data migration

When you finish data migration with success, don't forget to save it for future generations.

The best option is to prepare a report which includes all information about the migration. You should mention things like people who were involved in migration, their tasks, dates, and stages of the process in the report. This way, you'll be able to quickly find out what's been done, for instance a year later when someone finally finds a little mistake to repair.

Building migration capabilities

Assigning dedicated resources to data migration is recommended in organizations that face legacy, siloed, and redundant systems on the technical side, or mergers, acquisitions, and partnerships involving data exchange on the business side.

Since data migration shares requirements for technologies and personnel skills with data integration, consolidation, quality, system upgrades, and enterprise data architecture projects, the investment should gain efficiency by addressing all these in a common organizational structure. That structure could be tightly focused, like an integration competency centre.

Read more about hopp tech and migFx on www.hopp.tech or contact us at info@hopp.tech.