WHITE PAPER



Everything You Need to Know About Complex Data Migration

Challenges and best practices

INTRODUCTION

Data migration is a complex process. It needs a robust methodology and good supporting software to deliver a high-quality result as quickly and cost-efficiently as possible.

These tips will help you identify the challenges and risks in any data migration project, and show that there are better and worse ways to handle them. Moving data can be complicated and time-consuming, but if you prepare well and think the process through, you can avoid the common errors.

Data migration has come a long way, and you should look for tools that automate as much of the process as possible.

01

Our common challenge

Data migrations have been around for as long as new systems or processes have replaced older ones.

Think of Gutenberg's printing press, when handwritten books had to be "migrated" to the new technology, or of medical records moving from paper to digital.

The examples are endless, and yet the industry has no common good practice. In many ways, data migration is an overlooked and immature corner of an otherwise advanced IT industry.

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

Good practice

Many best practices focus on the iterative nature of data migration: a continuous cycle of steps that includes profiling the legacy data, designing the target system, creating the transformations and setting up test cases.

Good practice involves end users in the development, and coordinates the migration with the program that rolls out the new system. It also keeps the old and new systems in step, hands the new platform over to its administrators, and retires the legacy platform.

Some of these tasks are business issues. They need collaboration between the IT migration team and the business users affected by the migrated systems.

Together, they make data migration a complex and challenging task, best done with a feature-rich tool that supports iterative methods.

Bad practice

Far too often, data migration projects lack a structured methodology, the right technology and clear goals, and are held back by tight budgets and strict deadlines.

Some argue that subject matter experts are unnecessary, because data migration looks like a straightforward transfer of data to a known system.

Data migration is often seen as a pure data task, left to the IT department and tacked on to the end of the much higher-profile implementation of the new system.

The new system may look good. But without the right people on the migration, looking good is all it will do.

02

Background

When a new system comes in, everyone involved, from managers to users, wants to know how their old data will move to the new setup, whether a vendor, consultants or their own IT team does the work. This paper discusses the important steps and questions in a data migration.

We aim to give clear, practical answers that are widely accepted in the industry. We also show where our approach is different, and why we believe it makes migrations better and more cost-effective.

> **Learn from others, and keep innovating**
>
> Our take on these challenges is an attempt to improve today's practice, which we believe can be much better. We built our approach into software, Hopp, as an alternative to the industry's current standard practice.

03

The challenge

Data migration is usually defined as the process of moving data from a legacy system to a new system.

It may sound like a simple exercise, but it often turns out to be harder in practice: many data migration projects partly fail on cost, time or quality.

The more complex the data and the logic behind it, the harder the migration. If you do not plan the mapping, validation and testing properly, the migration can go wrong, cost more, and even put the new system at risk.

There is no sure way to do it perfectly, but a step-by-step plan helps you avoid the common problems. Every project is different, yet we believe there is a common approach to data migration, and it is worth thinking about best practice so you can learn from the mistakes of others.

We are all trying to get better at data migration: smoother, cheaper and faster. This paper shares what we have learned, and suggests an approach that helps your project go well.

Think twice

Organizations' needs for system support change constantly. Successful data migrations help them keep up, with systems ready for new initiatives that benefit users and customers.

Data in itself has little value. What matters is how it is used. Organizations use their data to make decisions, deliver their products and services, support customers and remember what has happened.

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

For a migration to go well, you need to know how the data should be organized and used in the new system. Be careful not to lose important information on the way: think about what data is moved, and what is not. We call the unique structure and use of the data the business logic behind it.

> **Think before you start**
>
> Not knowing, or not caring about, the business logic can put the whole migration at risk, and leave users unable to do their work after go-live.

Remember that data migration is as much about people, customers and processes as it is about data and systems.

Data migration can be complex, and it needs good cooperation between people across the organization, business and technical. Keep communication open with everyone involved in, or affected by, the changes the project brings.

Some useful definitions

Data has to be moved in many situations, for different reasons:

- **Data consolidation or integration:** combining several data sources into one or a few, as in a data warehouse, usually with frequent updates.
- **Data migration:** moving data from one or more systems or data models to another, as in a one-off system change or a system upgrade.
- **Complexity:** by this we mean how different the source and target data models are, along with the business rules that go with them.

The main difference between integration and migration is how often, and why, the data, along with its business rules and functions, is moved to another system or data model.

Much of a migration project's time goes into specifying and building the maps and transformations between the source and target systems. The more different the systems are, in data models or business rules, the longer the project takes, across analysis, specification, development, testing and training.

Preparing a strategy, following best practice and avoiding common problems all raise the chance of success. We believe the lessons and good practices in this paper will help you get there.

04

The bigger picture

A moving target

Do not wait until everything is perfectly planned before you begin. Start as early as possible, because things usually take longer than expected. At the outset, do not wait for a complete technical specification of the target schema, functions and business processes. Later in the project, though, it matters a great deal whether the implementation team can keep the design changes under control.

Scope the project

This is not the place to rewrite the book on project management, but a data migration project, like any other, needs thorough planning, management and execution. So, a few words on project management before we move on.

Scoping should identify the issues and risks that may come up later, so the migration team can mitigate the known ones.

The aim of scoping is to define the project thoroughly, and let stakeholders review it before kickoff. We believe that review should cover the project's structure, approach, requirements and technical aspects. Focused on the core methodology, it should answer these questions:

- Which alternative methodologies have been considered?
- Are the chosen methodology and tools fit for purpose?
- Does the migration tool match the skills of the people assigned to the project?
- Is the team adequately staffed and skilled?
- Is there a data quality plan?
- What are the overall timeline and activities based on?
- Are the timelines realistic, given readiness, methodology, tools and resources?
- What are the technical features of the proposed migration tool?
- How is the success of the migration judged, and who signs off the data once it has been migrated?

These questions come on top of the usual challenges of any large, complex project: managing stakeholders, communication and change. A data migration adds several critical issues of methodology and tools, which most organizations have little experience with.

Have a plan

Data migration may not be the central focus of the overall plan, for example in a new system implementation, but its key activities need to be outlined and managed separately, so they get the attention they need.

Project initiation and training make sure everyone knows what is expected of them and can do the job, especially in data migration, 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 the tasks.

In most projects, it is hard to match people, skills and tasks. If you know what resources you need to get the job done, you will be well prepared.

Plan for the unknown

The tasks, people and skills required are closely linked to the choice of methodology. How do you make a credible plan for activities you are not familiar with, or have never tried? How realistic is your plan when you just don't know?

What usually happens is an overall milestone plan, with teams asked to meet the deadlines. Most project plans include target dates and timelines for team activities, but they often lack an overview of the workflows needed to reach those points.

Data migration projects are iterative and hard to plan in the smallest detail, but it is important to define the key milestones and dependencies as part of the overall program plan before the project begins. That clarity removes confusion once the project is under way, and it shows you the gaps in your resourcing, where you lack the skills or the budget.

How good is your plan?

If you have followed common practice, your project plan should have a reasonable level of detail. But planning without perfect knowledge means things may not go as expected. Several areas can surprise you in a data migration project:

- Data quality is not as expected.
- Activities take longer than expected.
- The system implementation side of the project generates new requirements.
- Required data is not available in the legacy systems.
- Changes in the overall program plan move the deadlines.
- People become unavailable or unhelpful, or are reassigned to other priorities.

For example, there is no point in setting an arbitrary four-week window for "data quality activities" if the data turns out to be truly abysmal.

You need to understand the dependencies in the data and business logic before you start mapping, and you cannot do that until the analysis and discovery phase is complete.

Do not rely on someone else's plan from a previous data migration. Your plan is set by the conditions of your own project, and of the wider program it sits in.

05

Choose a methodology

A well-suited methodology is an important part of a successful data migration. There is no standard industry-wide approach, so most organizations have their own methodology, or none at all.

The traditional approach is to hand-code the migration with a combination of scripts, SQL and homegrown routines, ideally based on detailed written mapping specifications from business analysts. This may seem reasonable and cost-effective, but there are good reasons to change it.

It is always hard to plan for the unknown, so take the time to understand the unknowns. The best way is a methodology that lets you try, test and learn quickly, in an iterative and incremental process, ideally supported by an easy-to-use tool.

> You cannot set accurate timelines before you have looked at the data, and at the complexity of the business logic to be migrated.

More recently, teams have adopted an agile approach to custom code development. This suits the iterative nature of data migration better, but it can lead to an unstructured codebase that is hard to change in later iterations.

If you write your own code from scratch, it may fit your needs better than something others have prepared. It would be your own, and highly tailored. But can you guarantee that it is free of bugs, and that everyone else can use it?

The best applications a company builds take thousands of hours to develop and test. Can you afford that for your own migration code? Probably not. So where you can, choose tools that already exist, and use the work someone else has done for you.

The right tools have already solved the problems every migration shares, so your team does not have to solve them again: tracing data lineage, version control, data quality, speed and complexity, and the loss of key people.

We believe the project approach and the methodology must work together, not against each other. Our methodology is a set of proven software tools and a project process designed to make full use of them. Together with proper project planning, it gives you what you need.

Project approach

Still thinking of a waterfall approach for your data migration? It is time to change that.

Data migrations do not suit a waterfall approach, yet most data migration projects still follow one.

An agile, step-by-step approach with focused deliveries is much more effective. Make sure your overall plan is flexible enough to handle the changes you expect, and leaves room to learn and improve as you go.

Design the project around iterations that each deliver a result, so that it moves forward in increments. Build rapid discovery into each iteration, and plan the project on what you discover, not on what you think you know.

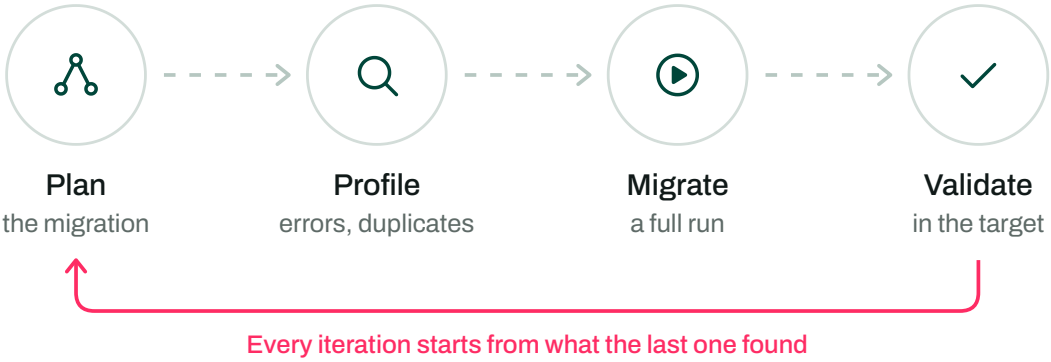


Figure 1. A data migration is a cycle, not a sequence: each run shows what the next plan has to cover.

Do not make the mistake of planning the entire project at the outset: you simply do not know enough yet.

Convince your stakeholders that an agile, transparent approach is the best way to manage risk and deliver a migration that benefits the business. And if you use a toolset to implement the project, it must support that agile approach.

06

Data quality and profiling

The big question here is whether you have the data the new system needs to work as intended. Is the data complete, and is it of good quality?

Expect the migration to uncover data quality issues. Some are problems that must be fixed; others are opportunities to enrich the data. Either way, set the bar high, and improve the data and metadata during the migration. If you don't, the project has a big challenge to overcome before it gets far.

The best way to tackle it

The business should know where all the current data is used, and what data the new system or process needs to work as intended. It should also know the shortcomings of the existing data, where it hurts today, and how the new system is expected to help. Identifying every data source and requirement early is vital, so that the migrated data lets the new system deliver the benefits identified.

At this stage, we find all the data, make it available with its known relationships, and look at it to understand the challenges it may hold. Work backward from the new system or process to identify the critical requirements. Never start with "this is our data, where do we want it to go?"

Once all the required data is collected, we need to understand how to transform it to meet the requirements of the new system. In an ideal world you would migrate everything, but in reality some data adds no value, or is covered by another source.

Use data quality rules to measure the quality of the data, and to fix or mitigate data quality problems. For each problem, decide how to reach acceptable data quality in the target system.

Once the data quality rules are in place, build them into the migration process, so they are measured on a data set in every iteration.

Options for fixing data quality issues

- **Ignore:** the issue is not significant for the migration, and can be fixed later in the target system.
- **Correct during the migration:** add a rule that transforms the data into a more manageable or final form. The drawback is a more complex transformation, and possibly a harder transition.
- **Correct in the source system:** usually the best solution, especially when the fix needs user input.
- **Correct in the target system:** the issue is postponed and fixed in the target system, as long as it does not affect the use of the new system.

07

Use tools, don't hand-code

Some developers still prefer hand-coding, although it is less productive. They can't break the habit, consultancies use it to bill more hours, and some managers put short-term savings ahead of long-term productivity.

For larger data migrations, there is no question: specifying, developing and deploying with a tool is far more productive than a hand-coded solution, because the tool already does the work every migration repeats, and a hand-coded solution builds it again each time.

Many solutions, tools and technologies are available for data migration, and more than one may be useful in a single project. Each migration has its own combination of existing systems and requirements for the new system. Choose tools on those factors, and on how often, and how deeply, you will do data migrations and similar projects in the future.

Tools for data migration fall into three groups:

1. **Database replication tools.** Easy and accessible, but they lack important features for complex data migrations. They often move data in one direction only, between instances of the same database brand, without transforming it.
2. **Extract, transform and load (ETL) tools.** Designed for data integration, but often used for data migration too. Organizations choose them for their ability to handle extreme requirements: large data sets, data transformations, deep data profiling and data integration.

3. **Tools built for data migration.** We built Hopp for the transformation in data migration. It has features that support the agile methodology discussed here, and a structure that helps manage and control the project.

08

Getting on with the job

We start by analyzing the requirements and quantifying the data gaps. The legacy systems may have issues, some already known to the end users.

Other problems are found only when the source data is examined for migration, or even later, when the data is inserted into the new system. Flush these issues out as early as possible, so the right action can be taken.

Detailed mapping

Mapping is often the most time-consuming phase of a data migration, and the most important. Here you define how each data element links from the source to the target system. All objects should be mapped down to the attribute level.

Traditionally, a business analyst does this and documents it in long written specifications, which are then passed to a developer to implement in an ETL-type tool or to code by hand.

Alternatively, the developer talks directly to the company's subject matter experts and prototypes the code over several iterations. Both routes are open to misunderstandings and ambiguities that may not surface until late in the process, or even after go-live.

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

Transforming legacy data is usually complex. Some source fields hold business logic that has to be identified and mapped separately. Legacy data may have missing values that must be filled from other sources. And there are the usual transformations for standardization and quality, as well as rules for conflict resolution, exceptions and branching.

So mappings are always complex, made of many parts that must be designed and tested individually before they are strung together in a data flow. Given that complexity, and the iterative design method, avoid hand-coding, and do this work in the interface of a dedicated tool.

In projects that do not use Hopp, the team managing the migration may not be the team that codes the migration logic.

> **We map differently**
>
> In Hopp, mapping and coding are one step. Business analysts map in Hopp Studio, and Hopp Core generates the code from their mappings. There is no back and forth between analyst and developer, and the mapped logic is never coded by hand.

Execution and flow

The mappings and rules that link the source and target systems must be published and coded accurately, and the right flows set up. That lets the execution team find the root cause of any issue found later. With several teams involved, from business analyst to developer to execution team, project management, communication and a common reference are crucial to progress and quality.

> **We execute differently**
>
> Hopp Core generates the code and runs it, and Hopp Portal controls the runs, down to a single business object. That makes testing, error handling and event management quick, and the whole process fast and interactive. There are no handoffs between analyst, developer and execution team to wait for, or to misunderstand.

Refine the mappings and validations in an iterative process, improving them until the final migration run. Ideally, you should be able to see and test the data in the target system in the final iterations.

Traditional waterfall models do not easily allow many iterations, which makes learning and continuous improvement hard.

Testing and tracking

Final testing is almost as important as the data itself. You have not succeeded until you have tested everything in the target system.

> **We test differently**
>
> Hopp Portal lets the business's subject matter experts review and investigate the issues found in the data, and see the source and target data at each stage to diagnose the problems. The results of every stage of the transformation are in Portal, a web application that everyone involved can open.

It is always better to give some testing tasks to real users, and to other people from the business. That gives the most accurate control and a proper evaluation.

It is common to wait until the job is done to validate the results, and sometimes it is the only option. But the sooner errors are caught, the easier they are to correct or to work around. So one of the best practices in data migration is to watch the migration while it runs. That saves a lot of time and, in the end, a lot of money.

In an ideal world, a data migration follows software engineering best practice, with separate environments for development, testing and deployment. In data migration, though, the deployment environment is usually the target application. And since development includes data profiling and the iterative testing of mappings and transformations, it often makes sense to combine development and testing in one environment.

Reporting progress

Make sure the reporting lets everyone involved see the progress, and the issues that affect them or need their response: a project manager who needs daily reports, an IT operator who wants to see current runtime issues, or a business leader who follows overall progress every week.

Interface specification

In our terminology, the interfaces needed are usually the ones that extract the data from your legacy systems, and the one that loads it into the target system.

09

After the data migration

When the migration is complete, keep a record of it for the future.

Write a report with everything about the migration: the people involved, their tasks, the dates and the stages of the process. Then, if someone finds a small bug a year later, you can quickly find out what was done.

Building migration capabilities

Dedicated data migration resources are worth having in organizations with legacy, siloed and redundant systems on the technical side, or with mergers, acquisitions and partnerships that exchange data on the business side.

Data migration shares technology and skills with data integration, consolidation, data quality, system upgrades and enterprise data architecture, so it is more efficient to handle them in one organizational structure, such as an integration competency center.

> **Want to know more?**
>
> Read more about Hopp at hopp.tech, or write to us at sales@hopp.tech.



hopp.tech

sales@hopp.tech

Hopp Tech Ltd
86-90 Paul Street
London EC2A 4NE
United Kingdom

Hopp Tech ApS
Gl. Kongevej 11, 3rd floor
1610 Copenhagen
Denmark