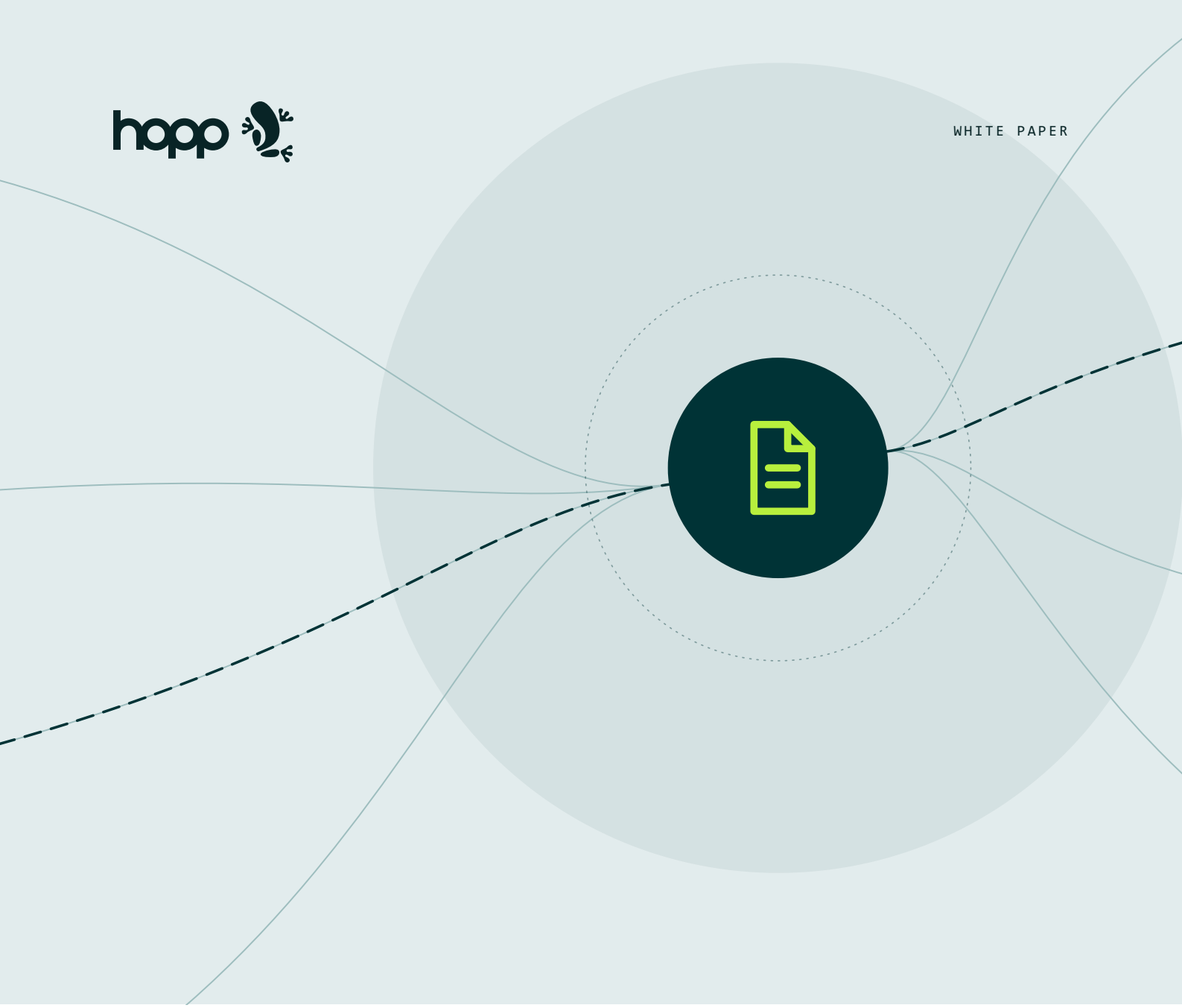
WHITE PAPER



An Agile Data Migration Approach

Why iterative delivery suits data migration, and how Hopp supports it

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. Agile data management is an iterative, step-by-step approach that puts great emphasis on continuous improvement, flexibility and team input.

Agile data management lets teams create business value quickly through continuous feedback, delivery and improvement in a changing landscape. It suits situations where traditional approaches do not give the flexibility needed and data requirements change rapidly.

This paper will help you decide the best strategy for your next data migration project.

01

Data migration methodology

In project management, a methodology is the blueprint that guides a project to success. Many are available, each with advantages and disadvantages, including the popular agile and waterfall approaches.

To avoid confusion, let's first define the two terms before going into more detail about each.

The waterfall methodology is the traditional approach to project management. It uses sequential phases to define, build, test and release project deliverables. Each phase is completed and approved before the team moves on to the next, and the project cannot go back to an earlier phase.

Agile methodology is an umbrella term for several more recent approaches that use iterative work cycles, often called sprints. Each sprint is a mini-phase that defines, builds, tests and releases a smaller deliverable.

The main difference lies in the goals. In the waterfall method, everything should be done right the first time; in the agile method, things should be completed quickly. Agile learns by trial and error; waterfall builds to a specification.

How the two differ on adaptability, testing, documentation and collaboration is less obvious. Put simply, the waterfall approach goes through its phases (design, development, testing and implementation) only once, while the agile approach goes through them many times over the life of the project.

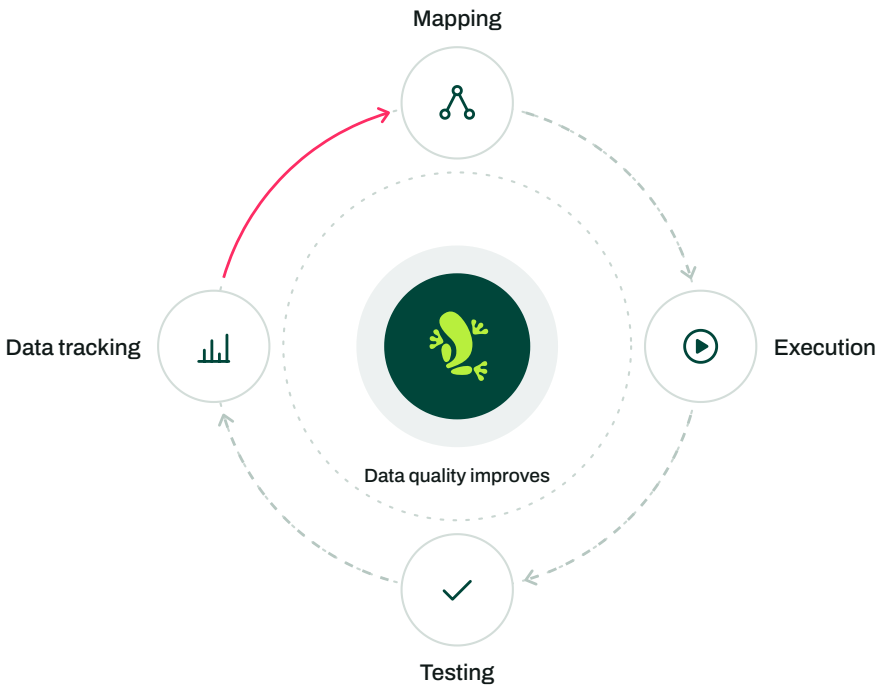


Figure 1. Every cycle maps, runs, tests and tracks the data, and each one starts from what the last one found.

The goal of most projects is to create value: new business opportunities, cost savings or better processes for the organization.

Figure 2 shows how the two methods differ in when they deliver that value. Agile development creates transparency, adaptability and value from the start of the process, and it reduces many of the risks along the way.

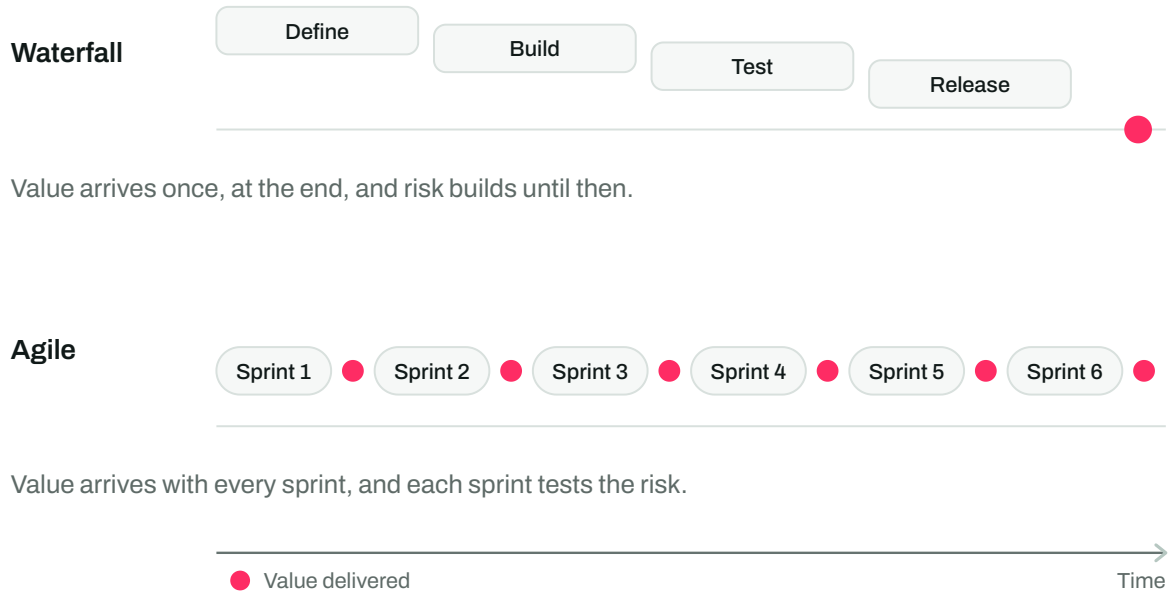


Figure 2. Waterfall delivers its value once, at the end. Agile delivers some with every sprint.

The growing focus on creating value quickly with minimal risk often moves projects toward a more agile approach. To reduce the risk of mistakes, lost time or value not delivered, you can use several test and release cycles rather than one big bang.

A word of caution: agile methods have broad support today, but they are not without challenges, especially when scaled to large enterprise projects. Putting sound principles into practice is often harder than expected.

02

Agile vs. waterfall

Let's look more closely at the features of agile that we believe are critical to creating value in data migration, in speed, cost and results:

- Adaptability
- Testing
- Collaboration

Through these three features, we look at the benefits and show why agile methods are the better choice for most data migration projects.

Adaptability

Adaptability is how responsive a project approach is, and how quickly it responds to change. A good example is how it handles requirements that change during the project.

The waterfall model is not adaptable, because its requirements are usually fixed and signed off at the start. Changes must go through a change request process, or not happen at all. Significant changes often mean the project has to start over.

Agile methods adapt to changing requirements, because the project reviews and validates the requirements at each step. Any change is simply reflected in the next step. This affects both the project team and the organization's stakeholders.

The waterfall method needs detailed documentation at each step to capture the project's scope and requirements, and implementation means reading the documents and carrying out each step.

Agile methods focus on releasing work quickly, so documentation is usually written in parallel, later, or never. With little information available up front, it is harder to give decision-makers and stakeholders an overview of the project.

Testing

Testing verifies that a deliverable does what it is supposed to do and delivers the expected value.

In a waterfall project, the deliverables are usually brought together toward the end, with testing as the final approval. You do not know whether the deliverable works until the project is almost done, and an error in any part can send the project back to the start, which costs time and money.

In an agile project, parts of the deliverable are tested throughout. If one component fails, it is easier to fix, because only that part is tested and revised. Fixing a small part costs less time and money, so the project as a whole is less affected.

Collaboration

In the waterfall approach, collaboration with stakeholders happens during the business specification at the start of the project, and again at the final data delivery. In the agile approach, collaboration is continuous, for the whole of the project.

03

Why agile development for data migration?

Let's now apply these features to a data migration project, and see why agile is the better way to tackle a complex, large-scale migration.

Large change efforts that involve new systems and processes are usually organized as a program: a series of dependent projects that deliver the changes, and their value, to the stakeholders.

In most programs, the data migration project has to start before the design phase of the main development project is complete. Otherwise, the final data migration solution is ready only after the main solution has been tested. To avoid that delay, the migration starts with an incomplete requirements specification, for example before the detailed design of the target system is finished.

The data migration team also usually delivers several intermediate deliverables, to meet the overall plan as well as the final delivery. Meeting these interim deliverables while requirements trickle in from the main solution design makes the migration project complex, and hard to estimate and plan.

This is why the tried and trusted waterfall approach does not suit a data migration project. A prototyping or agile approach suits it best.

The agile approach requires the sprints to follow an overall vision and feature backlog. It also requires the development to fit an overall architecture that automates the delivery as far as possible. Hopp provides that architecture, and much of the automation, so project teams can work in an agile way during the migration.

Each sprint should deliver usable migrated data that supports the development and test schedule of the main program. That way the migration project can work in an agile way even when the main program follows a waterfall approach.

The user stories should focus on the details of the extract and transformation rules, while the solution's architecture is already in place and gives a stable environment for developing those rules. Hopp's architecture lets you progress in an agile way without having to redo much of the solution later.

Working within a defined architecture is what makes it possible to start early, before the final design of the target is known: you know what the end result will look like, even if not in detail.

Done badly, agile development can lead to an unstructured tangle of code. Hopp's structure prevents this, while still allowing rapid iterations and the execution of a partly developed solution.

Figure 3 shows the overall scope of a migration as full coverage of the business objects, across, and full depth of their attributes and quality, down.

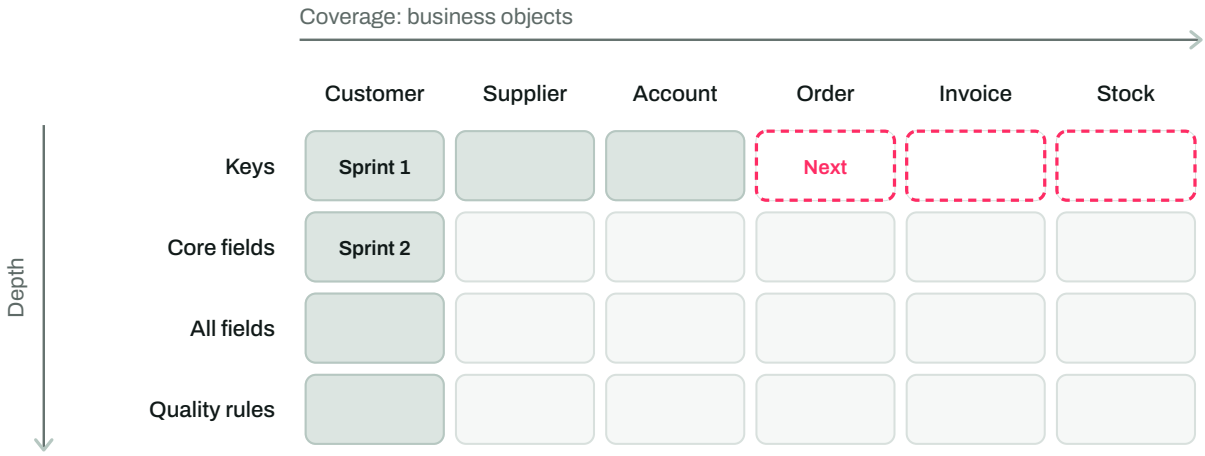


Figure 3. A sprint can go across, a few fields of many objects, or down, every field of one. Static data such as customers comes before the dynamic data that depends on it.

A sprint can then cover a few attributes across several business objects, or many attributes of a single business object, filling in the whole scope step by step. The two can be combined, so that every attribute of a few objects is complete before all the objects have been started. This flexibility lets the project continue before the target design is complete, or while some legacy systems are still undocumented.

Data falls into two broad groups: slow-moving or static data, and fast-moving or dynamic data. Customers, suppliers, accounts and people are static data; purchase orders, invoices, card transactions and stock levels are dynamic. Some static data usually has to be migrated before dynamic data can be, and these dependencies must be taken into account when choosing what goes into each sprint.

04

How Hopp supports agile

Hopp builds the dependencies into the definitions, so it can show where data or specifications are missing that would stop the sample data from being processed. Finding these gaps early in the sprint, while there is still time to act, keeps the sprint from failing.

Collaboration in Hopp

Another critical part of the agile approach is cross-functional teams. Technical and business people both take part in the development, so the outcome meets every business requirement.

Hopp supports this by letting the business's subject matter experts take an active part: at the start, by defining requirements, and at the end, by reviewing the results and correcting them where needed.

- **Hopp Studio** records the target and source data structures and their content, along with the business rules that govern them, in a structured way and in a place the subject matter experts can use. Functional and technical consultants then define there the transformations that turn source data into a form the target accepts.
- **Hopp Core** generates the executable code from what was specified in Studio, and runs it on real source data to produce data that can be tested in the target system.
- **Hopp Portal** controls those runs and keeps an audit trail of every one.

Testing with Hopp

Hopp Portal lets subject matter experts and functional consultants review the results of each run and decide what to correct when errors are found. It also lets the project manager review the results and produce detailed reports for the rest of the program management.

05

In summary

The typical data migration project begins without a firm foundation, and its structure has to be built as the project goes. Starting without a structure leads to difficulties and delays later, and in some cases a complete rework.

Hopp provides a structured environment in which the agile approach works for data migration. Without it, the development team would have to design and build that environment before productive work could begin.

> **Want a demo?**
>
> To find out more about Hopp, or to see how it supports an agile migration on your own systems, 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