
An agile Approach to 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.*

The agile method implies an iterative and incremental method of management. It focuses on helping teams in an evolving landscape to maintain focus on rapid delivery of business value. Agile is based on continuous improvement, flexibility, input of the team, and delivery of results with high quality.

We hope that this paper will help you when considering the best way to approach your next data migration project – and of course use our data migration toolkit - migFx - for it.

A word on methodology

Project management methodologies are like the blueprint of a project, thoughts to guide the project team to a successful completion of its task.

There are many different methodologies out there - all with their pros and cons. Some are labelled "agile" and other "waterfall" methods.

To be clear on the terminology used in this paper we start with a small recap to get on the same page.

The **waterfall method** is the traditional project management approach that 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 phase. The project can't move backwards to previous phases.

Agile methods are an umbrella term covering several newer project management approaches that use different forms of iterative work cycles, sometimes called sprints. Each sprint is a 'mini-phase' used to define, build, test, and release the smaller project deliverables.

The main difference between waterfall and agile methods is in the goals; the waterfall method wants to get everything right the first time, and agile methods want to get things released quickly.

Learning through trial and error are features of the agile mind while build to specification is the waterfall way.

Differences are visible in way the different approaches lend themselves to adaptability,



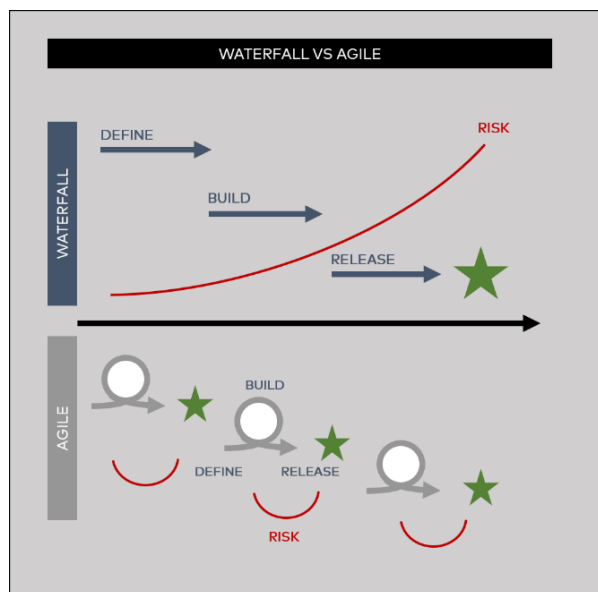
documentation, testing, and collaboration in support of the different goals.

Let's put it this way. Waterfall move through different stages or phases like design, develop and test, implement only once while the agile approach have multiple such iterations over the project life cycle.

The objective of most projects is typically to deliver "value" like improvements in form of new business opportunities, cost savings or improved processes to the organisation.

The figure below illustrates the difference regarding the value proposition of both methodologies and how agile development tends to deliver visibility, adaptability and value in the beginning of the process and reduces a lot the risks during the project.

Figure 1: Process, Value and Risk dynamics.



The increasing focus on delivering value fast with a minimum of risk often drive projects towards a more agile approach in project management.

To minimise the risk of getting something wrong, spending too much time or failing to deliver the "value" can be better managed if more learning can be done through multiple testing and release cycles and not by a big bang.

A word of caution. Agile methods have a lot of support today but are not without their challenges specially when scaling to large enterprise projects. Turning sound principles into practice is often harder than expected.

A closer look at agile vs waterfall

Let's zoom in on the different key features of agile that we find crucial in delivering the value in data migration - faster, cheaper and better.

- *Adaptability*
- *Testing*
- *Collaboration*

These three features help us discuss the benefits and to argue that agile methods are the better choice for most data migration projects.

Let's start discussing the different features and compare them in waterfall vs agile methods.

Adaptability is concerned with how responsive the project approach is - just how quickly it reacts to changes throughout the project.

A good example is how to manage changing requirements during the project cycle.

Waterfall is not adaptable as they are typically set out and signed off initially. Changes need to go through change requirements process or not at all. Very often, larger changes will force the project to start all over.

Agile methods are very adaptable to changing requirements because the project reviews and validates the requirements at every step throughout the project. So, any change in requirements is simply addressed during the next step.

This has implications for the project team as well as the organisation (stakeholders).

The waterfall method requires detailed documentation at each step, to understand



the project scope and requirements is a matter of reading the documents.

Agile methods are more focused on quickly releasing the deliverables, so documentation tends to be written in parallel, after or never. Very little information is available as we go or before, so it would be harder to provide an overview of the project to decision makers and stakeholders.

Testing is verifying the deliverable does what it's supposed to and that that are linked to the value expected.

In a waterfall project, the different deliverables are typically tied together towards the end with testing being the ultimate approval. So, you won't know if the deliverable works until you're nearly done with the project. Any error in any part of deliverable will send the project back to the start to make a change, which will impact cost and time.

On the other side, agile projects test parts of the deliverable throughout the project. So, if one part fails, it can be more easily resolved because you're only testing and redoing one part of the deliverable. Fixing a small part takes less time and money, so the overall project is impacted less.

Collaboration, describe the way the different methods assume that you are working with other team members and stakeholders.

Through collaboration, the project knows if the stakeholder is happy with the deliverables as the project proceed. Collaboration is front-loaded in waterfall projects and an ongoing process in agile projects.

Why Agile Development for data migration?

Now let's apply the features of agile vs waterfall to a data migration project and see why agile is the only way to approach any complex, large-scale data migration.

Big change efforts in any organisation involving system and process changes are typically organised in a project program. A project program is number of dependent projects designed to deliver the change – or value to the stakeholders.

In most project programmes there is a requirement to start the data migration project before the design phase of the main development project is complete, otherwise the final data migration solution will not be ready until after the testing of the main solution.

To avoid this delay, it is necessary to start the data migration project with an incomplete requirement statement, i.e. the detailed design of the target system is not finalised.

In addition, typically the data migration team will have a number of intermediate deliverables to meet the overall plan's requirements as well as the final deliverable at Cutover.

The task of meeting these interim deliverables with the overall drip feed of requirements from the main solution design makes the data migration project complex and difficult to estimate and plan conventionally.

The "tried and trusted" waterfall approach to project planning is therefore unsuited to a data migration project. A prototyping or agile approach is what is suited for data migrations projects.

The agile approach requires that the iterations (sprints) of development align with an overall vision and a feature backlog, it also requires that the development of code fits within an overall architecture in order that the delivery is automated as far as possible.

migFx provides that architecture and a large part of the automation to enable the data migration project teams to operate in an agile manner.

The features should be aimed at delivering usable instances of migrated data that



support the development and testing schedule of the main programme, in this way an agile approach can be used for the data migration project even if the main programme is working on a waterfall approach.

The user stories should be aimed at developing the detail of the extract and transformation rules, the architecture of the solution should be in place and provide a stable environment for that agile development of the rules.

The migFx toolset provide such an architecture, one that will allow you to progress in an agile way without ending up having to redo a large part of the solution at a later stage.

Working within a defined architecture best supports the "start early before the final design of the target is known" approach.

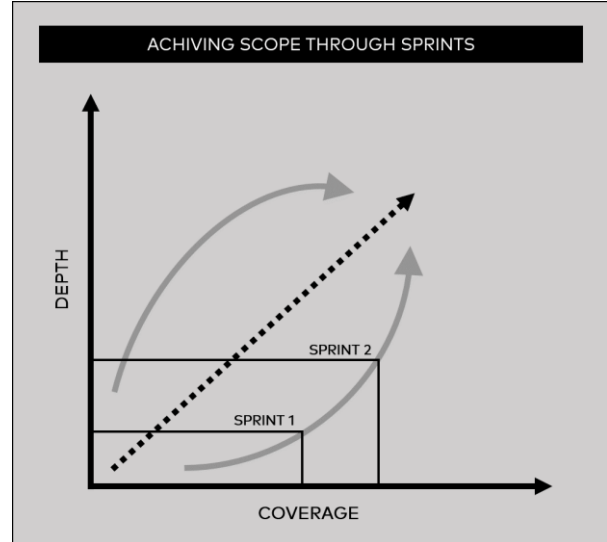
For the start early approach to work you need to work within a designed architecture, so you know what the outline of the final result will be if not the detail.

Agile development, if not implemented correctly, can lead to an unstructured tangle of code. The structure imposed by migFx prevents this while still allowing fast iteration of development and execution of a partially developed solution.

If the overall scope of a migration is viewed as achieving full business object "coverage" and full "depth" in attributes and quality we can then illustrate this as in Figure 2 by having coverage along the x axis and their depth along they axis.

The sprints in an agile development can then be defined as a small number of attributes across a number of business objects or a larger number of attributes across a single business object to gradually fill out the whole scope.

Figure 2: Sprint through coverage and depth



These different approaches can be combined so that all the attributes for a small number of objects are completed before even all of the objects have been started.

This flexibility allows the project to proceed before the target design is completed or where not all the legacy systems have been documented.

Data can be split into two broad categories, slow moving or static data and fast moving or dynamic data. Examples of static data are Customer, Supplier, Account, Person, while examples of dynamic data are Purchase Order, Invoice, Credit Card Transaction, Stock Levels.

It is often necessary to migrate an amount of static data before dynamic data can be migrated, and these dependencies must be taken into account when selecting the content of the sprints in an agile development.

migFx supports agile

When using migFx, these dependencies can be built into migFx definitions so that the product identifies where there is missing data or specifications that would prevent the processing of the sample data provided.

This can save a significant amount of time in identifying gaps early in the sprint while there



is still time to take action and prevent the failure of not delivering the sprint.

Collaboration using migFx

Another key element of the agile approach is the building of cross-functional teams. It is important that both technical and business resources are applied to the development process so that the result satisfies all requirements.

migFx supports this by providing a mix of functionality that enables the business subject matter experts to actively participate in the development process at both beginning by defining the requirement right through to the end by reviewing and if necessary correcting the results.

The Studio component of the tool allows the structured recording of both Target and Source data structures and content, together with the business rules that govern them, in an environment accessible to the subject matter experts.

It then enables the functional and technical consultants to define the transformations necessary to process data from the source into a form that is acceptable to the target.

The Engine component enables the rapid production of executable code from the definitions recorded in the Studio to enable the processing of real source data into a form that can be tested in the target system, the code is controlled in this execution by the Director which also maintains an audit trail of the execution runs.

Testing with migFx

Finally, the Tracker enables both the subject matter experts and the functional consultants to review the results of the transformation process and decide what corrective action is required where errors are detected. It also enables the project manager to review the results of the processing to produce detailed

and statistically sound reporting to communicate to the remainder of the programme management.

Summary

The typical data migration project starts with no firm foundation. The structure will gradually have to be established and build during the project. Starting without a structure cause difficulties and delays further down the line and - in some cases - the complete rework.

migFx provides a structured environment which enables the effective application of the agile approach to data migration projects, which otherwise would have to be designed and built by the development team before any productive work can begin.

Thanks to [John Smith](#) for inspiration and co-drafting this paper with us.

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