white paper

WhereScape®

Traditional vs. Automated Data Warehouse Development

Part 1: The Traditional Approach

The image below shows the traditional data warehouse development process, from requirements through to logical and physical modelling, architecture, storage and index management, ETL mapping, deployment and eventually maintenance and documentation. Each step has its own tool and its own team, and usually takes from between six weeks and six months or more to complete. Once you get to the end it is very complicated to make retrospective changes.

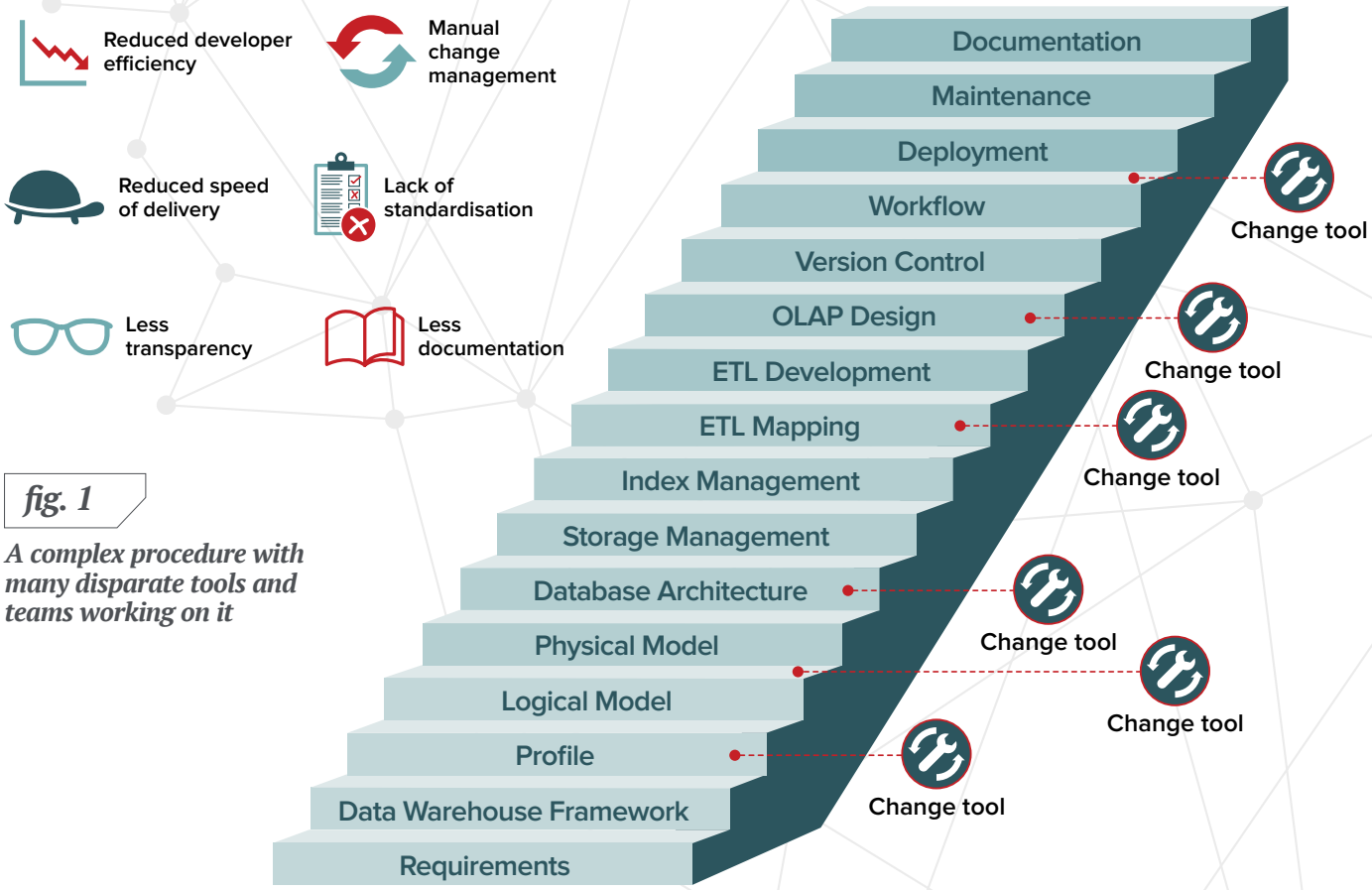


fig. 1

A complex procedure with many disparate tools and teams working on it

1.1 Requirements and Modeling

Business requirements form the foundation of the technical staircase. Once all requirements are collected, the logical model, or reference architecture as it is known, is created. This tests whether the data warehouse should work in theory and is usually presented informally in Microsoft Word, Excel or Powerpoint. Then we need the data itself. We need to understand its sources, where they are located, what quality the data is in and how much of it there is. Then it's time for the profiling and modelling, which is traditionally where tools like Erwin or Power Designer are used. Models are built to understand the source system, the core data warehouse and target data marts. Using conventional tools and methodologies, this process takes weeks and months. With WhereScape you can get here days and weeks.

1.2 Physicalizing the Model

Once the model is approved, it needs to be physicalized. Typically the following questions would be asked:

- **What schemas and databases are we going to use?**
- **What protocols are we going to use to transfer data?**
- **What security access are we going to use?**
- **How are we going to store the data: multi-parallel? Column-wise? Row-wise?**
- **Where do we need indexing and what kind of indexing are we going to use?**
- **Where are the bottlenecks and what are we going to do to resolve them?**

This is all usually done with database tools: Oracle TOAD, Microsoft SQL Server Management Studio or DB Architect to name a few. Many of these are proprietary to the chosen database, so could be locked into a certain toolset.

Once these questions have been answered, the code to move data from A to B to C and so on needs to be written. This is traditionally done with ETL mapping and development tools such as Informatica, DataStage or Microsoft SSIS. These tools move the data from the data sources to a landing area, transform that into a staging area, then move it into data models such as star schemas, data vaults and so on. This is all done with platform specific code such as SQL, Python or C Sharp. All this work needs version control so you can go back to old versions if something goes wrong, which often involves another tool such as GitHub.

1.3 End User Interaction

After this development work is complete, the question of how end users are going to interact with the data must be considered. Things like raw data vaults are very storage-friendly but not user-friendly. So teams tend to build data marts such as star schemas with facts and dimensions, and use them as presentation access layers, often with an OLAP cube on top so requests can be processed rapidly.

Then the workflows need to be chosen: how often does the data warehouse need to be refresh? Daily? Weekly? By the second? By the minute? Real-time? How often do we need to do it to keep the business informed and up-to-date? A workflow tool might be used here such as IBM ClearCase, and a scheduler such as Airflow, Control-M or Maestro.

1.4 Pushing into Production and Making Changes

Once all the above has been achieved in a development environment, the infrastructure and needs duplicating in a test environment. When testing is finished, it is pushed into a production environment to make sure everything is exactly the same. This deployment process uses another tool such as IBM ClearQuest or JIRA.

Once the model is deployed after three to nine months of work to this point, there is the issue of maintenance. Every time a change is needed because of a new requirement, or perhaps something didn't perform as it should have in testing, we start again from the bottom of the staircase and follow the same process:

- What are the requirements?
- Check the source
- Model the source
- Look at the physical infrastructure required
- Are there any changes to table structures?
- Do we need more storage or compute space?
- Do we need more network bandwidth?
- How is the ETL affected?
- Is an extra data needed in the pipelines?
- If so, how does this affect the scalability of the flows?
- Move into the test environment
- Move into the production environment

1.5 Project Success and Maintenance

Data Warehouses that take this long to build often fall down precisely because they took that long. The requirements might have changed completely in that time as the business evolves. Technology can move on drastically at a rapid pace, so the original design could be outdated or defunct. Then there is the day-to-day maintenance. When a data warehouse has been running for a few years, it develops issues that need to be managed.

Along the foundation of the staircase in figure 1 is the word governance. This is the idea that the whole process is being tracked, controlled and quantified at management level, traditionally by the CIO but perhaps these days by the CDO or CTO. With all the different tools and teams in the stairway process, it's extremely difficult for a single person, or even a small committee, to have control.

1.6 Communication and Human Error

At each step step on figure 1 you are changing people and tools. After one team has finished with the code it passes it onto the next, which can often lead to human error and confusion if the completed work and instructions are not explained properly or sufficiently documented. Each team only sees what the last team sends them; they are often not aware of the original requirements and contextual conversations between the teams that precede them.

Requirements from the business are given to the framework people. They produce some architecture and pass it over to the modellers, then it is passed onto the builders who do the coding and send it to the testers and so on. It's likely the new team won't be ready to start working on the project right away, causing further delays. Some of these processes overlap, but it tends to be a linear path as Figure 1 suggests, with little loops back along the way.

When the stairway methodology was in its heyday in the late 90s and early 00s, offshoring was very popular as a lot of staff were needed for such a complex project. Companies would send their instructions to offshore countries with cheaper labour, and often be disappointed when the work came back. They would then work out that often their requirements or documentation were not correct. Teams were disparate enough when they were in the same building, so when you added in teams in different countries, on different time zones, the flow of the project got even more disjointed, inefficient and time-consuming.

Part 2: Why Take the Stairs When You Can Take the Escalator?

Since the heyday of the traditional approach in the early 00s, the attitude towards software engineering has changed dramatically with the proliferation of the Agile Manifesto:

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right,
we value the items on the left more.

**taken from agilemanifesto.org*

In practice, the elements of this manifesto most relevant to data warehousing have been customer collaboration and the ability to respond to changes in customer requests (the customer in this instance being the business user rather than an external customer). With the scale of data warehousing projects, often automation is the only way to respond at a speed close to that which is now demanded.

"With WhereScape, our design and development is highly automated. The low total cost of ownership and the return on investment of 300 percent more than justifies our investment in WhereScape."

"We couldn't have done all this with other tools. Wherescape made it possible for us to leverage a small budget and take it to where we are today."

Su Rayburn,
AVP of Information Management and Analytics, Delta Community Credit Union

2.1 The Switch to Agile

With this in mind, the process in Figure 1 should be largely redundant today, but in reality only a small, albeit growing percentage are actually doing Agile development while many large companies continue with the traditional model.

Over the past couple of years there has been a real shift towards using Data Automation as the backbone of any serious data warehousing project. This is how it works.

2.2 Automated Scoping and Modeling

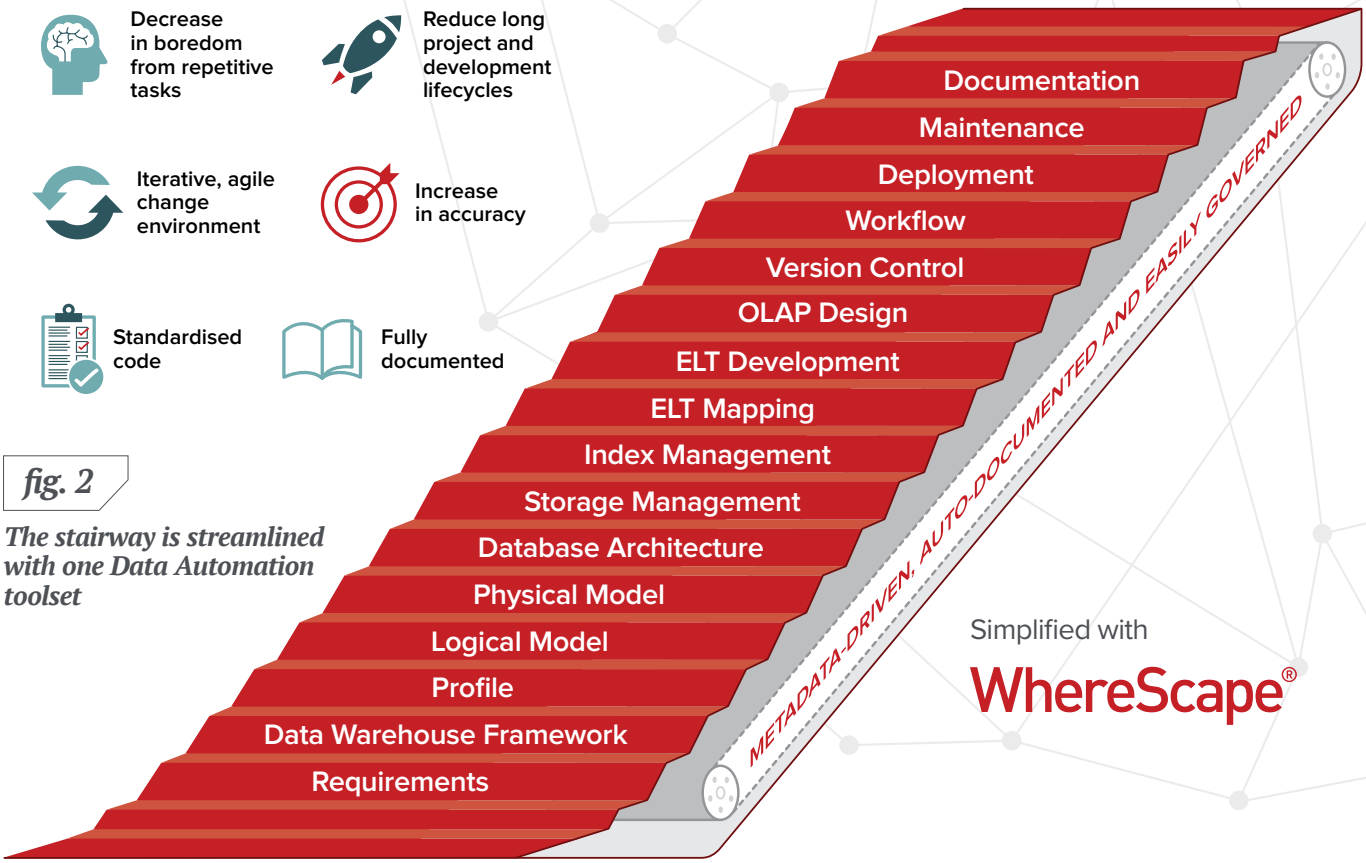


fig. 2

The stairway is streamlined with one Data Automation toolset

In Figure 2 above, the whole process and all its steps are managed in one toolset/framework and performed by the same team (often the same person). Rather than a staircase, the process is more of an escalator. Using a common software means every act is taken according to the same standards and conventions.

Once connected to your data ecosystem, WhereScape quickly scopes all data sources to give a profile of the data available, its quality, location and so on. Prototype architectures can then be spun up using this data in under an hour, and show it to the colleague(s) who requested it and check all requirements are understood before the build starts with no misunderstandings or misinterpretations. With the traditional approach, the process would be that the modelling team takes the requirements and the requestor only sees the finished article days or weeks later six months later or longer when the project has reached the top of the staircase and changes are difficult to implement.

2.3 ELT Code Generation

Once the model has been approved by all parties, the code that physicalizes it is automatically generated in WhereScape's Integrated Development Environment. Rather than ETL developers having to spend months writing all the code by hand, WhereScape writes thousands of lines of SQL code in seconds and uses ELT instead of ETL, so the transformation work is done by the power of the database, not the intermediary 'black box' used by legacy ETL tools. The code is native to your existing platform as WhereScape is database agnostic. It merely sits on top of the infrastructure you already have and makes it perform much faster using metadata.

Time-consuming elements of the traditional approach, such as storage and index management, mapping and version control are done automatically by WhereScape, saving months of intricate work and difficult decision-making processes that take up a lot of person hours. Those long lists of questions in section 1 are largely irrelevant because WhereScape makes these decisions according to industry best practice templates. However, it is important to point out that these templates are configurable if required, so the developers maintain control throughout. The resulting architecture is tested and pushed into production.

2.4 Automated Documentation and Change Management

The whole process above can be instantly documented in at the click of a button. This means the whole architecture is described in a common language, and there is no delay in waiting for developers to get round to doing a job that is time-consuming and often left until last or not done at all. If any of the developers were unavailable or had left the company, someone could read the documentation, so the knowledge is not locked away in the minds of a handful of tech leads as may be the case in many companies.

With a modern, automated architecture, the CDO can provide governance for the entire process because he/she can track everything in WhereScape, with full lineage and automatically generated documentation. Every action taken is tracked and timestamped. As the majority of actions are automated, once the CDO sees that the tool can be trusted, they only really need to keep track of the elements along the stairway that are controlled by human decision-making. The direction and outcome of the project still lie in human control of course, but the undertaking of the time-consuming, repetitive tasks is performed by the tool for added reliability and shorter projects.

2.5 The Speed of Data Automation

The real wow factor with WhereScape comes when we do a proof of concept at a prospect company, and **our techies are able to complete the whole process in figure 2 in just three to five days.** We often say conservatively that our tool increases the output of each developer by 5x, but when figure 1 can take between six months to multiple years this figure is often tens or hundreds of times higher. This is why we have such a high success rate when we get to the proof of concept stage and show what the tool can do. These gains in speed are life-changing for data teams who have been working under the traditional methodology , and this actually changes working lives, not just project completion dates.

Once the automated building process is complete, changes are much faster to commit as the tool can either pinpoint and tweak the relevant section of metadata and make all necessary up-and down-stream alterations to avoid unexpected repercussions, or it can simply rebuild the element you want to change from scratch given the minimal time and effort required to do so. Changes and bug fixes can now be completed in two hours rather than the days weeks or months required when multiple teams have used multiple tools to shape the code over a long period of time.

2.6 Future-Proofing Your Architecture

WhereScape is metadata-driven and version managed. This means there is a reliable record of the whole architecture and its processes kept in a repository at all times, and so all this can be more easily migrated onto a different platform, Cloud or otherwise, without the need to replicate all the work you have already done.

Many companies use the adoption of WhereScape as a good excuse to switch to a Cloud database they have been wanting to use for some time, because WhereScape can generate the huge amount of code needed to migrate existing data and data infrastructure from one platform to another.

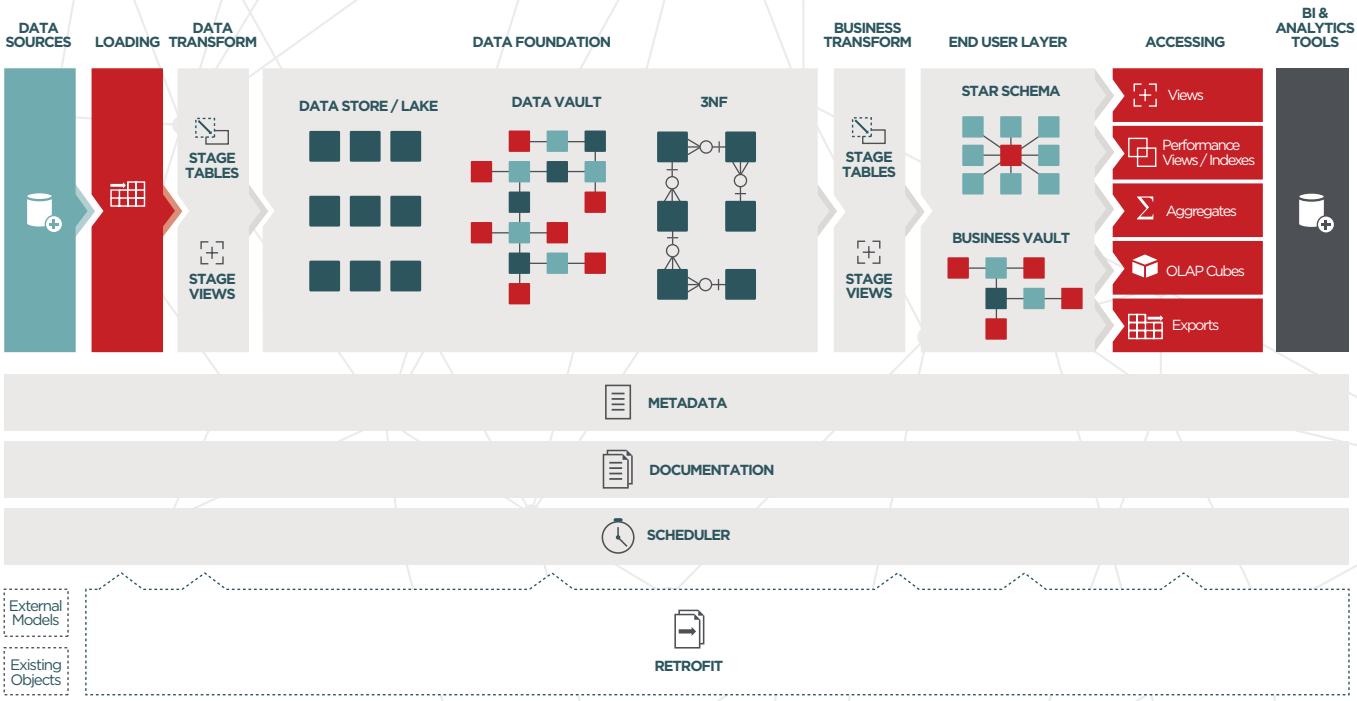
"Union Bank used WhereScape RED to rapidly scope, prototype, and build out its business facing data mart from source systems in less than eight weeks. Using conventional development methods would have required six to eight months.

WhereScape RED is providing Union Bank a very powerful, repeatable, low-cost way to develop in parallel and with agility. We're seeing probably a 5x performance increase in developer productivity."

Ryan Fenner,
MUFG Union Bank's VP, Data Solutions Architect.

Others opt to automate their existing architecture, safe in the knowledge that if they choose to switch in the future, they can do so easily and are not locked into any provider. This means that the choice of database is no longer a ten-year decision as it used to be. In fact, WhereScape customers such as Legal & General opt to use WhereScape as a common data warehousing build and management tool, and allow each of their departments to work on the database that suits their needs best.

Once organizations realize the incredible savings in time and resources that are possible, they quickly recognize the possibilities available to data teams using WhereScape in the age of automation.



To see how WhereScape works for yourself, please register for one of our in-depth, 60-minute Data Automation Demos, on the first Thursday of every month.

wherescape.com

"With the last project we had 10-15 external resources, and now with automation we have 90% of the project staffed internally, with some help from [WhereScape partner] IT-Logix for staff augmentation. The idea is that our internal people can really become the masters and drive the whole process. That wasn't possible before WhereScape. How much money do we save? It's in the millions."

Gallus Messmer,
Data Warehouse Architect at Swiss insurance company Helsana.