

4. DELIVERY PROCESS

A data warehouse is never static; it evolves as the business expands. As the business evolves, its requirements keep changing and therefore a data warehouse must be designed to ride with these changes. Hence a data warehouse system needs to be flexible.

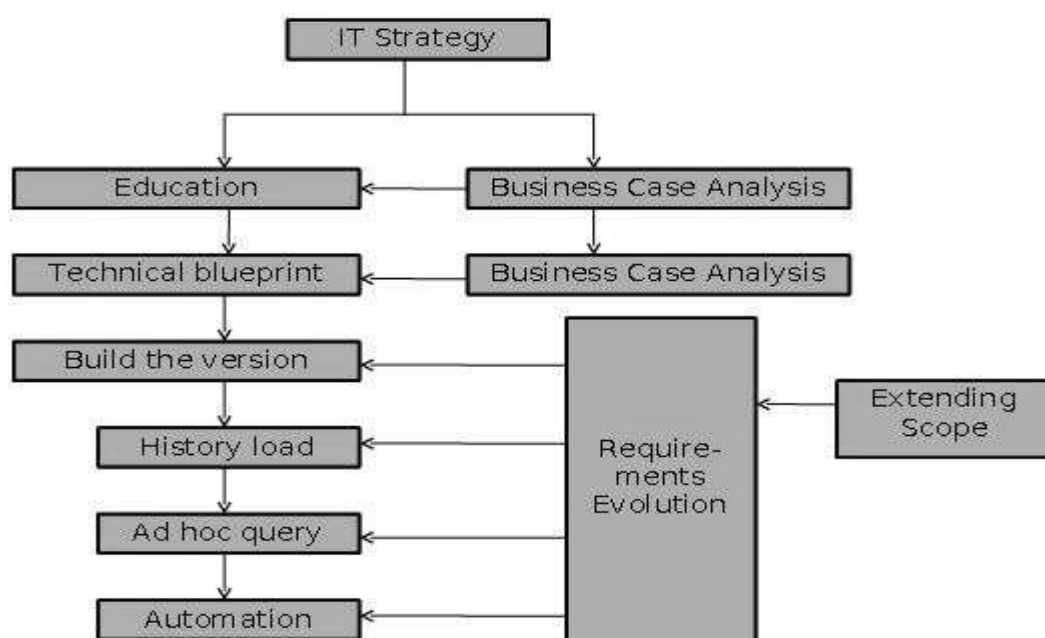
Ideally there should be a delivery process to deliver a data warehouse. However data warehouse projects normally suffer from various issues that make it difficult to complete tasks and deliverables in the strict and ordered fashion demanded by the waterfall method. Most of the times, the requirements are not understood completely. The architectures, designs, and build components can be completed only after gathering and studying all the requirements.

Delivery Method

The delivery method is a variant of the joint application development approach adopted for the delivery of a data warehouse. We have staged the data warehouse delivery process to minimize risks. The approach that we will discuss here does not reduce the overall delivery time-scales but ensures the business benefits are delivered incrementally through the development process.

Note: The delivery process is broken into phases to reduce the project and delivery risk.

The following diagram explains the stages in the delivery process:



IT Strategy

Data warehouses are strategic investments that require a business process to generate benefits. IT Strategy is required to procure and retain funding for the project.

Business Case

The objective of business case is to estimate business benefits that should be derived from using a data warehouse. These benefits may not be quantifiable but the projected benefits need to be clearly stated. If a data warehouse does not have a clear business case, then the business tends to suffer from credibility problems at some stage during the delivery process. Therefore in data warehouse projects, we need to understand the business case for investment.

Education and Prototyping

Organizations experiment with the concept of data analysis and educate themselves on the value of having a data warehouse before settling for a solution. This is addressed by prototyping. It helps in understanding the feasibility and benefits of a data warehouse. The prototyping activity on a small scale can promote educational process as long as:

- ☐ The prototype addresses a defined technical objective.
- ☐ The prototype can be thrown away after the feasibility concept has been shown.
- ☐ The activity addresses a small subset of eventual data content of the data warehouse.
- ☐ The activity timescale is non-critical.

The following points are to be kept in mind to produce an early release and deliver business benefits.

- ☐ Identify the architecture that is capable of evolving.
- ☐ Focus on business requirements and technical blueprint phases.
- ☐ Limit the scope of the first build phase to the minimum that delivers business benefits.
- ☐ Understand the short-term and medium-term requirements of the data warehouse.

Business Requirements

To provide quality deliverables, we should make sure the overall requirements are understood. If we understand the business requirements for both short-term

and medium-term, then we can design a solution to fulfil short-term requirements. The short-term solution can then be grown to a full solution.

The following aspects are determined in this stage:

- ☐ The business rule to be applied on data.
- ☐ The logical model for information within the data warehouse.
- ☐ The query profiles for the immediate requirement.
- ☐ The source systems that provide this data.

Technical Blueprint

This phase needs to deliver an overall architecture satisfying the long-term requirements. This phase also delivers the components that must be implemented on a short-term basis to derive any business benefit. A blueprint identifies the following:

- ☐ The overall system architecture.
- ☐ The data retention policy.
- ☐ The backup and recovery strategy.
- ☐ The server and data mart architecture.
- ☐ The capacity plan for hardware and infrastructure.
- ☐ The components of database design.

Building the Version

In this stage, the first production deliverable is produced. This production deliverable is the smallest component of a data warehouse. This smallest component adds business benefit.

History Load

This is the phase where the remainder of the required history is loaded into the data warehouse. In this phase, we do not add any new entities, but additional physical tables would probably be created to store increased data volumes.

Let us take an example. Suppose the build version phase has delivered a retail sales analysis data warehouse with 2 months' worth of history. This information will allow the user to analyze only the recent trends and address the short-term issues. The user in this case cannot identify annual and seasonal trends. To help him do so, last 2 years' sales history could be loaded from the archive. Now the 40GB data is extended to 400GB.

Note: The backup and recovery procedures may become complex, therefore it is recommended to perform this activity within a separate phase.

Ad hoc Query

In this phase, we configure an ad hoc query tool that is used to operate a data warehouse. These tools can generate the database query.

Note: It is recommended not to use these access tools when the database is being substantially modified.

Automation

In this phase, operational management processes are fully automated. These would include:

- ☐ Transforming the data into a form suitable for analysis.
- ☐ Monitoring query profiles and determining appropriate aggregations to maintain system performance.
- ☐ Extracting and loading data from different source systems.
- ☐ Generating aggregations from predefined definitions within the data warehouse.
- ☐ Backing up, restoring, and archiving the data.

Extending Scope

In this phase, the data warehouse is extended to address a new set of business requirements. The scope can be extended in two ways:

- ☐ By loading additional data into the data warehouse.
- ☐ By introducing new data marts using the existing information.

Note: This phase should be performed separately, since it involves substantial efforts and complexity.

Requirements Evolution

From the perspective of delivery process, the requirements are always changeable. They are not static. The delivery process must support this and allow these changes to be reflected within the system.

This issue is addressed by designing the data warehouse around the use of data within business processes, as opposed to the data requirements of existing queries.

The architecture is designed to change and grow to match the business needs, the process operates as a pseudo-application development process, where the new requirements are continually fed into the development activities and the partial deliverables are produced. These partial deliverables are fed back to the users and then reworked ensuring that the overall system is continually updated to meet the business needs.