

White Paper

Iterative Data Warehouse Development

Methodology: A Strategy to Demonstrate Real Value and Minimize Risks

By Scott Moshier, PMP

Overview

While the concept that a data warehouse creates value by enabling data-driven business decisions as been adopted as a best practice, the development of data warehouses is still prone to failure or budget overruns. This paper asserts that the development methodology greatly affects the risk of failure. Further, by implementing an iterative methodology that is focused on justifiable business cases throughout the lifecycle, a data warehouse project can prove its value very early in the development process.

There is a large body of literature debating the “correct” architecture for a data warehouse. The work of Ralph Kimball and Bill Inmon are the two most popular approaches. It is the belief of *Real Results Consulting* that the appropriate approach to a data warehouse architecture will vary depending on the organization which it supports. Therefore, this paper does not identify any particular architecture as “correct”; but rather, attempts to address how a project team should approach development once an appropriate architecture has been identified.

Data Warehouse Development Challenges

While all projects have risks, data warehouse development poses unique challenges due to its scope. Many levels of management and diverse business areas will use the data warehouse to analyze many complex issues. This is quite different from implementing a transactional system that automates one or a group of defined business processes. Therefore, it is necessary to consider the challenging components unique to a data warehouse project:

- **Political Will** – Since one of the high-level goals for a data warehouse is to provide an integrated view of the entire business (e.g., integrating finance, marketing, operations, etc...), it is of significant value to executive management. Therefore, the project needs a very high-level business sponsor whose support for the project can cut through organizational politics and is clearly evident to all business units. The sponsor also needs to

- ensure that adequate resources are made available and a clear business case justifies the investment. Simply put, the organization needs the political will to appropriately initiate and fund a data warehouse project.
- **Engaging Business Unit Leaders** – Closely related to an organization's political will is the ability to effectively engage business unit leaders to ensure effective data integration. Since it is likely that many transactional systems are managed by individual business units, actively engaging those people who are responsible for the transactional systems is of key importance. These are the people who will know both the history and idiosyncrasies of their systems and will make the planning and analysis of a data warehouse effective. A key component for engaging business unit leaders is proving the value of a data warehouse to them. This involves understanding their issues and challenges and clearly demonstrating how the data warehouse will help them.
 - **Top to Bottom Integration** – Another unique challenge of a data warehouse is the need for tight integration from high-level business goals all the way down to database fields. While this may seem obvious, many data warehouse projects start development without clear direction regarding those high-level business goals. For example: a data warehouse project with only the vague goal of being able to identify cost savings could go in many different directions without more clear guidance. In addition to the high-level business goals to guide the project, there is a large amount of work needed to integrate very different sets of transactional data (e.g., operational vs. marketing data). Due to the complexity of the integration, it is easy to spent a great deal of time in the planning and analysis phase. This leads to the last point below.
 - **Time Creates Risk** – The longer the project team takes to perform planning and analysis tasks, the more risk it creates for the project. It is understood that a successful project needs adequate planning and analysis to create the right "road map to success". However, the longer the project team takes to show the value of a data warehouse to its stakeholders, the more skeptical the stakeholders become of the return on investment (ROI) and the ability of the project team to get the job done.

Given that the above factors are just *some* of the challenges involved in data warehouse development, the sponsor and project manager need to actively manage these risks. The sponsor can "clear a path"

and give the project the clout it needs to succeed. However, it is then up to the project manager to create a team that has the best chance of success. Most of the best practice information today focuses on the hardware and software involved as well as the skills of the various team members. However, little has been said about a very important aspect of the development process – a development methodology. The next sections describe why this is important and highlights common processes for any data warehouse project.

The Methodology Matters

Even with strong executive support, experienced developers and appropriate hardware/software, a data warehouse implementation can still get mired in endless analysis and development. Consider a traditional development approach where the team: analyzes requirements, designs, develops, tests and then finally deploys the solution. In this model, the development team interacts with the stakeholders at the beginning and then “go away” to develop the solution. They do not return to the stakeholders until they test the solution (or sometimes not until deployment). While this is conceptually easy to understand and implement, it rarely reflects the real world where requirements change during the design, development or test phases. This model also makes change later in the process very expensive because the entire solution has already been designed and developed based on old, now invalid information. Therefore, the developers will have to re-design and re-develop to create a new solution. This can take significant time and, therefore, money. In addition to the extra money involved in re-work, the extra time to modify the original design can make executives and business unit leaders less confident in the team’s ability to do the job and anxious either to see ROI or end the project.

This scenario illustrates both the risks of a data warehouse implementation and the need for a development methodology that steadily moves the project along. To mitigate the risks, an effective development methodology needs to incorporate the following:

- **Flexibility** – The project team needs to be able to adjust to a changing environment. The methodology has to assume from the beginning that requirements will change and build in processes and controls to minimize the expense of changes.
- **Demonstrate Value** – To continue to have the support of executives and business leaders, these stakeholders need to see valuable results throughout the development lifecycle and not just at the end. Therefore, the methodology needs to

- incorporate processes where stakeholders regularly see the progress of the project.
- **Know When Changes Occur** – To keep up with changing requirements, it is necessary to keep in constant contact with stakeholders. Therefore a successful methodology ensures that developers and stakeholders regularly interact and verify (or change) the requirements.

To implement an effective methodology to minimize risks and create real value for stakeholders very early in the development process, *Real Results Consulting* recommends an iterative development methodology coupled with sound project management practices. While there are a variety of valid methodologies in use today, we recommend the use of the Rational Unified Process (RUP) concepts for a development methodology and the Project Management Institute's (PMI) framework for project management processes. During the initial phase of a project the methodology is customized to the particular project and environment. However, there are some commonalities for any data warehouse project. These are described below.

Methodology Implementation Highlights

Wherever you stand on the "Ralph Kimball vs. Bill Inmon" debate of data warehouse architectures, one thing is still certain: After all the debate, planning and analysis, the development work will begin and team needs clear understanding of what will be done and when. Therefore, solid project management practices need to be in place and an experienced project manager leading the effort. While project management processes will not guarantee a successful project, some processes like the Project Management Institute's (PMI) need to be incorporated to ensure that nothing "falls through the cracks". By developing the processes to support the PMI principles will make it more likely that all stakeholders will know what to expect and risks will be identified early in the planning process. Detailed information on the PMI processes can be found in their publication, "A Guide to the Project Management Body of Knowledge Third Edition (PMBOK Guide)". Related to the data warehouse development process, the PMI processes are invaluable in running the overall project (e.g., setting up cost monitoring or change management processes). However, since the PMI principles are general and not aimed specifically at IT development, it is necessary to supplement the overall project management methodology with a specific development methodology.

To minimize the risks and incorporate the best practices outlined above, an iterative development methodology needs to be employed.

By focusing on developing a sub-set of the requirements, stakeholders can validate (or change) requirements earlier in the process before an entire solution is developed. Additionally, it demonstrates the value of the project early in the lifecycle. Further, the shorter development cycle ensures that developers and stakeholders keep in closer contact throughout the entire project. There are several iterative methodologies currently used today (e.g., the Rational Unified Process (RUP), Extreme Programming (XP), SCRUM, etc...). Of these, *Real Results Consulting* recommends the Rational Unified Process (RUP) for its flexibility and ability to be tailored to the specifics of the project.

The Rational Unified Process

Those who only have a passing knowledge of RUP may think of it as only collecting use-cases or a bureaucratic methodology that can only be implemented with expensive IBM software. Fortunately, neither of these extremes are true. While use-cases are part of the methodology, there are many additional best practices included. While IBM does sell software specifically created to support the RUP process (e.g., Rational Rose, Rational RequisitePro, etc...), it is not necessary to own the software to implement the methodology.

RUP is a iterative process framework that has defined processes and roles, but is intended to be tailored to each unique project. There are workflows and disciplines that are designed to allow business cases to justify each successive iteration. While this paper cannot go into all the details of implementing RUP, below are the four phases of an iteration and some typical examples of what may be included.

- **Inception** – This first phase requires that a business case and success factors be analyzed in context of the core requirements, constraints and key features. Stakeholders must concur on the core requirements and it must be justified by the business case. Otherwise, the scope, features or business case must be rethought to avoid project failure in later phases. Examples of artifacts (i.e., documents) created during this phase may be:
 - ...an enterprise architecture document describing the current environment and any planned changes (e.g., purchasing new hardware or software).
 - ...a business case and success factors describing the benefits being sought from the project. This needs to be fairly detailed so that it can be both quantified and broken down into several iterations.
 - ...some document showing concurrence of the scope by stakeholders.

- **Elaboration** – The problem domain analysis is done and basic architecture takes form. During this phase, a stable architecture and vision of the solution must be demonstrated. That is, it is necessary to ensure that an organization’s infrastructure is appropriate for the project and that there is a concurrence on the vision of the end product. Of course the entire solution will not be designed at this point, but it is necessary to demonstrate the viability of the vision and the ability to add to the core functionality with each iteration. Examples of artifacts created during this phase may include:
 - ...a use-case model from which the core and additional requirements can be derived.
 - ...an overall project plan.
 - ...a design document describing the architecture and approach to implementing the core and additional features.

- **Construction** – This when the coding of the current iteration of the solution takes place. The main focus of this phase is to develop the core functionality that will serve as the foundation of additional iterations. Examples of artifacts created during this phase may include:
 - ...monitoring documents showing the progress of the development and adherence to the initial requirements.
 - ...quality and testing documents for this iteration.
 - ...change requests for subsequent iterations.

- **Transition** - In this phase, the solution is moved from the development organization to the end user. Depending on the iteration, the product may not be the entire solution. In this case, the development is complete to a pre-defined level of acceptability and then moved to users for initial training or beta-testing to validate against known requirements. Examples of artifacts created during this phase may include:
 - ...completed validation documents from the end-users.
 - ...change requests for subsequent iterations.
 - ...training documentation.

Using this methodology to manage the development process continues by requiring a business case justifying each iteration. Therefore, any changes can be incorporated at the appropriate time and the

functionality developed can be tightly integrated with the business needs (as outlined in the business case).

Conclusion

It may be possible to create a data warehouse without the methodologies outlined above. However with so many risks inherent in data warehouse development, all project teams need as many chances for success as possible. Industry experience has shown that the PMI and RUP methodologies can enhance the project and give the greatest chance of success. Of course, just implementing a methodology itself does nothing if it is not used to achieve the goals of moving the project forward and being flexible to changes in the environment. Therefore, the organization looking to implement these methodologies should not be expecting some pre-selected templates that can be a "silver bullet" for their project. Rather, this will create processes to effectively manage the risks and changes that would make other projects fail.

About the Author:

Scott Moshier has been leading IT projects for over eight years and is certified as a Project Management Professional (PMP). He has successfully implemented data warehousing strategies in a variety of industries including: travel, financial services and government. He is also the managing partner of the Indianapolis-based IT consulting firm, Real Results Consulting, LLC. In addition to leading out IT projects, Scott graduated from Purdue University and has an operations background where he re-engineered business processes to increase efficiency of business processes.