

**timeXtender®**



Achieving Rapid Return On Data Warehouse Investments

# Achieving Rapid Return On Data Warehouse Investments

## Introduction

Data warehouses are at the heart of most business intelligence solution platforms or frameworks and are expected to remain the foundation for relational reporting, OLAP analysis, in-memory analysis, and cloud-based business intelligence. Bill Inmon and Ralph Kimball are commonly considered the fathers of data warehousing. They represent very different philosophies and approaches to data warehousing and the *Inmon vs. Kimball* battle has been going on for years.

Inmon coined the term data warehouse and was the first to write about data warehousing. He defined the data warehouse as "...a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process."

Kimball is by far the most widely adopted of the two and a world renowned advocate of the dimensional approach to data warehousing. You will find his data warehousing toolkit books on the shelf of virtually any data warehousing professional.

## Top-down design

The Inmon approach is often called the top-down design, as it defines a centralized repository for the entire enterprise: the enterprise data warehouse (EDW). Data marts are subsets of information from the data warehouse and serve specific topics or departmental needs. An organization that takes this approach also takes on a large project that is very broad in scope and - not least - results in significant up-front costs.

## Bottom-up design

Kimball advocates an incremental staged approach where data marts are created for specific business processes. A data mart models a specific business process such as Sales or Inventory. When data marts consisting of dimensions and facts are integrated by means of the data warehouse bus architecture, they make up a data warehouse. It is an important aspect of the Kimball lifecycle that it starts with a comprehensive enterprise-wide analysis of the relevant business processes and the first data mart is therefore only built after the completion of this analysis phase.

## Microsoft SQL Server

Neither the Inmon nor the Kimball approach is specific to any database vendor, however, the specific focus of this whitepaper are data warehouse implementations on Microsoft SQL Server.

## Problem Statement

A data warehouse often deserves its reputation of being complex, time consuming and expensive to build and maintain. There are countless projects that have failed to deliver on time and budget and according to business requirements. Consequently, many organizations are reluctant to invest further in their data warehousing solutions.

One of the factors that contribute to the failing of data warehouses is the lack of realization that a data warehouse is not a one off project that solves all future business requirements. Another factor is failing to realize that time is one of the key ingredients of successful implementations. If you can afford to wait 6-7 months for Business Intelligence to support a business requirement, is it even worth doing?

### Key definitions

- Business Intelligence (BI) refers to analysis and reporting of data stored in a data warehouse
- Data warehouse (DW) is the foundation for Business Intelligence
- Extract, transform and load (ETL) refers to the process of extracting, cleansing, and conforming source data and loading it into the data warehouse

### Volatile business environment

Today's business environment is unquestionably highly volatile across most industries. Business requirements that change often demand rapid response from the business intelligence solution, which in turn drives changes to the data warehouse. However, with conventional project management methodologies and tools, business requirements are often obsolete before they are implemented in the data warehouse.

### Time to return on investment

Therefore, as stated above, time is a critical factor for any organization wanting to make well-informed decisions on a trustworthy foundation. The current economic climate adds further stress to most BI initiatives, as resources are limited and a fast return on investment is mandatory.

## Previous Options

The wide adoption of the Kimball methodology is often explained by higher flexibility and faster first results compared to the Inmon methodology. However, the major downside of both methodologies is that the duration of time from the start of a project to the point where end users begin to reap the benefits of the data warehouse is significant. Typically, the first results cannot be expected in less than 6-7 months.

### Requirements gathering

The conventional ways of gathering business requirements in data warehouse projects are very formalized and rely heavily on written documentation. This is apparently done out of the belief that the business will get what it needs by signing off on a business requirements definition document. In reality the business will *get what it wrote*, which is often different from what they need.

This is further amplified by the timing of the requirements gathering that commonly takes place in the months-long up-front analysis phase that covers the entire enterprise. When the requirements are finally implemented in the data warehouse months later, the business discovers that its needs have already changed.

The prolonged requirements gathering phase is the single biggest reason for the slow implementation cycles of conventional methodologies. The current toolset for designing, developing and deploying a data warehouse is another contributing factor. First, there is a big gap in tools for developers and information workers; second, the developer tools used for hard coded development are time consuming to learn and use.

## timeXtender Solution

The Kimball dimensional modeling approach is by nature incremental and starts with a single data mart, which is the model of a single business process. Data warehouses are built by adding additional data marts that are integrated by means of conformed dimensions and facts and by using the data warehouse bus architecture.

The Kimball dimensional modeling approach is robust, scalable, and flexible enough to deal with future requirements. The Kimball project lifecycle identifies task sequences and concurrent activities, but tends to be inherently slow as a result of the up-front business requirements definition task.

However, when the Kimball dimensional modeling approach is combined with artifacts and principles from agile software development methodologies, it leads to enterprise class data warehouses that can be delivered in short increments. This in turn ensures a fast return on investment.

### Solution strategy

1. Replace the extensive, enterprise-wide up front analysis with an incremental analysis approach that acknowledge that business requirements change frequently.
2. Work in short 1-2 weeks cycles where it is always the most important and current business requirements that are delivered.
3. Use a rapid design and deployment tool that automates code generation and documentation, and thereby enables the short cycled delivery strategy.

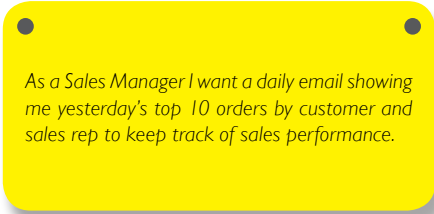
### User stories for agile requirements gathering

User stories are one of the key artifacts of SCRUM and Extreme Programming (XP) which are both agile software development methodologies. A user story is a high level definition of a business requirement and contains just enough details to be used for prioritization, planning, and communication between the business and the technical team. One way to think of a user story is using the 3 Cs:

- Card            User stories are written on cards.
- Conversation    The actual requirement is communicated from the business to developer through conversation.
- Confirmation    The acceptance test for the story.

User stories are written by the business in business language and terms because it is the business that needs to understand the requirements and be able to confirm when they are completely delivered by developers. User stories are well suited for data warehousing since you can reduce the time spent on up-front analysis significantly.

The sample user story below clearly contains enough information to prioritize this requirement among a set of requirements, even though the details are not fleshed out on the card. When this user story is selected for implementation, it is used as a starting point for a conversation between the business and the developer concerning the details of the requirement.



*As a Sales Manager I want a daily email showing me yesterday's top 10 orders by customer and sales rep to keep track of sales performance.*

Figure 1: A user story

A user story should be limited enough in scope that it can be implemented within 1-2 days and still deliver business value. The user story above specifies requirements both towards the BI application (delivered through email) and a data mart centered around a sales order fact table in the data warehouse. This is easily done within the 1-2 days, which makes the length of the story ideal.

#### Use prototyping and interact with the business

Regardless of whether you are using a conventional text-intensive requirements specification approach or high-level definitions like user stories, the actual requirement details are worked out through conversations between the business and development. When the actual model is delivered, the business often suggests modifications, which can then be implemented as a natural part of the development cycle.

While you follow the Kimball four step dimensional modeling approach, you build a dimensional model using manually entered sample data to demonstrate your interpretation of the requirements to the business. Next, you give the business user to a feel for the type of information the dimensional model can deliver by creating a quick front-end view in, for example, Microsoft Excel or PowerPivot.

The Kimball four step process:

1. Select the business process that you want to model. The business process is given by the user story you are working on.
2. Define the grain of the business process. In the example above, the grain might be defined as "an individual sales order line".
3. Choose the dimensions that apply to each fact table row. In the example above, dimensions are customer, sales reps, and calendar.
4. Identify the numeric facts (measures). This is an example of details that are not clearly specified in the user story, but will be worked out in the conversation with the business and confirmed using prototyping. (Are we measuring top 10 orders by quantity or amount? Are amount exclusive or inclusive of taxes, in which currency etc.)

The prototyping approach is a very powerful way to achieve final confirmation from the business on the dimensional model before entering the more expensive ETL process. The ETL process is known to be the most expensive part of BI and represents around 70% of the risk and effort of a BI project. Imagine the consequences in costs and delays of going through this expensive phase and delivering clean and conformed data into a dimensional model that in the end does not properly address the business requirements.

### Deliver early, deliver often, and work in small teams

Although the journey in itself can be enlightening, the value a data warehouse initiative is only delivered, when the business requirements are met. That is, when the data warehouse is able to deliver timely, accurate, and relevant information to business users to support decision making. Only at this point, will the business users begin to experience the promised benefits.

When the SCRUM approach is transferred to data warehouse implementation, the following assumptions are made.

- Design and development of a data warehouse can be a complex and unpredictable process.
- Not all requirements are known at the beginning of a process.
- Requirements can change during the process.

The most efficient way to deal with this is to work in short iterations, called *sprints* in SCRUM. A sprint is a time-boxed period of time during which the development team commits to delivering a number of business requirements that are communicated between the business and development as user stories.

Every sprint is expected to result in a fully working and tested model. This means the business can expect requirements to be delivered within the recommended sprint duration of 1-2 weeks.

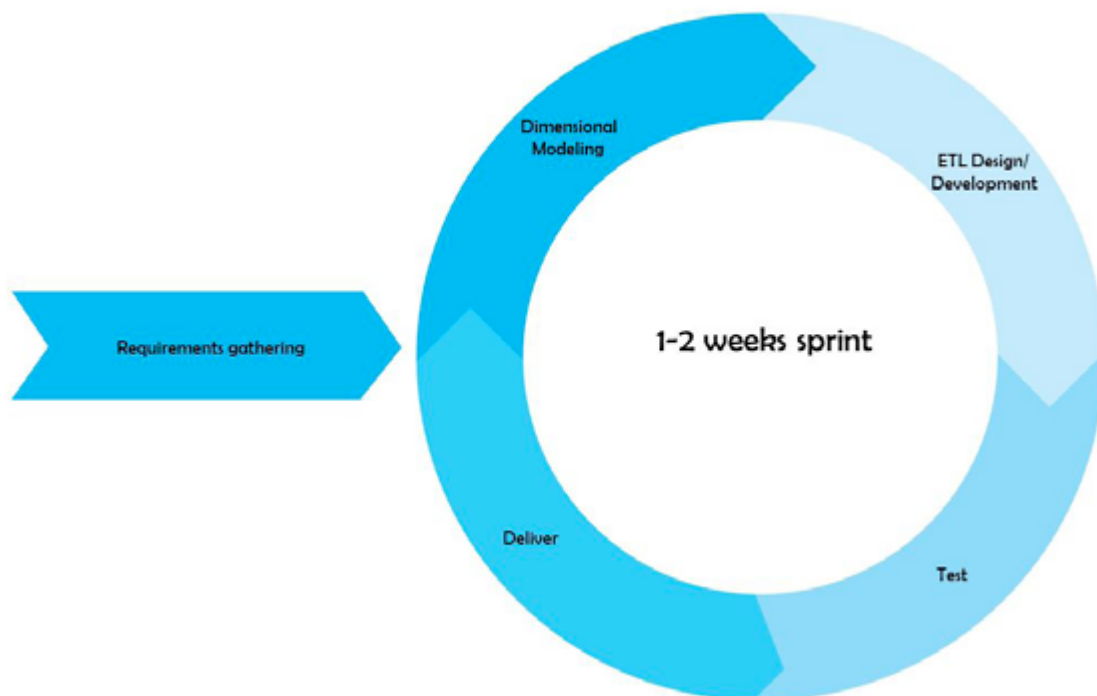


Figure 2 Sprint cycle

## Conclusion

### Implement business requirements based on business value

The success of any project depends on a solid understanding of the business requirements to successfully translate them into design considerations for the dimensional model. It is important to acknowledge that business requirements change frequently, and you should therefore always focus on the business requirements that add the most immediate business value. There is no point in spending weeks or months mapping out the details of business requirements that hold little value. It only serves to delay the ability to deliver the key requirements.

Every organization and project is different, and it is difficult to generally quantify the cost to benefit ratio; however, it is assumed that the benefits exceed the costs; otherwise it would add little value to the business.

The diagrams below illustrate when you can start harvesting the benefits of an agile, incremental approach. You will see the first benefits in less than a month compared to conventional approaches that will not show any benefits until after 6-7 months.

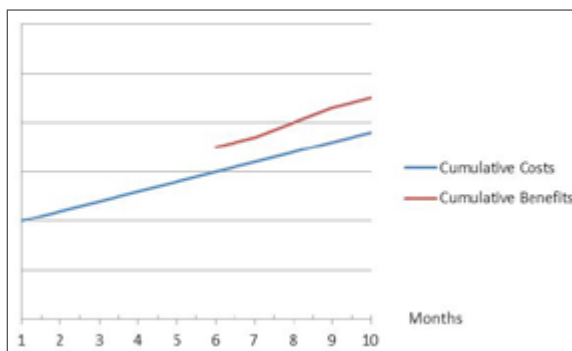


Figure 3 Conventional approach

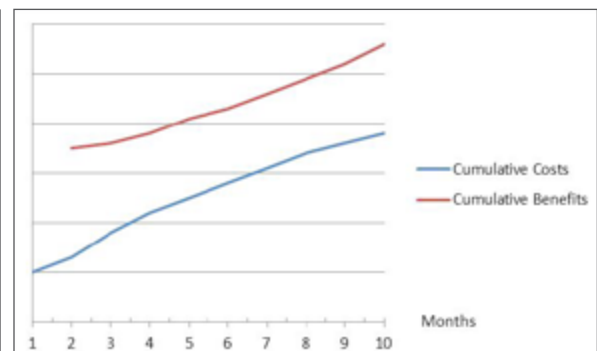


Figure 4 Agile, incremental approach

## References

The resources listed below provide more in depth information about data warehousing.

Organization	Address
Corporate Information Factory (Bill Inmon)	<a href="http://www.inmoncif.com">http://www.inmoncif.com</a>
Kimball Group (Ralph Kimball)	<a href="http://www.kimballgroup.com">http://www.kimballgroup.com</a>
The Data Warehousing Institute (TDWI)	<a href="http://www.tdwi.org">http://www.tdwi.org</a>

### Books

User Stories Applied: For Agile Software Development, Mike Cohn, ISBN: 0-321-20568-5

#### Disclaimer

*This document is provided for informational purposes only, and is subject to change without notice. The information contained in this document represents the current view of timeXtender on the issues discussed as of the date of publication. This document should not be interpreted to be a commitment on the part of timeXtender, and timeXtender cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only. timeXtender makes no warranties, express, implied, or statutory, with respect to the information in this document. Complying with all applicable copyright laws is the responsibility of the user. No part of this document may be reproduced or transmitted in any form without written permission from timeXtender.*

#### Copyright

*timeXtender® is a registered trademark of timeXtender Holding. Microsoft®, Microsoft® Dynamics®, and SQL Server® are either registered trademarks or trademarks of Microsoft Corporation, or Microsoft Business Solutions ApS in the United States and/or other countries. All other product names mentioned in this document may be trademarks or registered trademarks of their respective companies.*

© 2011 timeXtender as. All rights reserved.