CHAPTER 1

Introducing the Kimball Lifecycle

Before delving into the specifics of data warehouse/business intelligence (DW/BI) design, development, and deployment, we want to first introduce the Kimball Lifecycle methodology. The Kimball Lifecycle provides the overall framework that ties together the various activities of a DW/BI implementation. The Lifecycle also ties together the content of this book, setting the stage and providing context for the detailed information that unfolds in the subsequent chapters.

This chapter begins with a historical perspective on the origination and evolution of the Kimball Lifecycle. We introduce the Lifecycle roadmap, describing the major tasks and general guidelines for effectively using the Lifecycle throughout your project. Finally, we review the core vocabulary used in the book.

We recommend that all readers take the time to peruse this brief introductory chapter, even if you are involved in only one facet of the DW/BI project. We believe it is beneficial for the entire team to understand and visualize the big picture and overall game plan. This chapter focuses on the forest; each remaining chapter will turn its attention to the individual trees.

Lifecycle History Lesson

The Kimball Lifecycle methodology first took root at Metaphor Computer Systems in the 1980s. Metaphor was a pioneering decision support vendor; its hardware/software product offering was based on LAN technology with a relational database server and graphical user interface client built on a 32-bit operating system. Nearly a quarter century ago, analysts in large corporations were using Metaphor to build queries and download results into spreadsheets and graphs. Sounds familiar, doesn't it?

Most of this book's authors worked together to implement decision support solutions during the early days at Metaphor. At the time, there were no industry best practices or formal methodologies. But the sequential steps of decision support were as obvious then as they are now; our 1984 training manual described them as *extract, query, analysis*, and *presentation*.

The authors and other Metaphor colleagues began honing techniques and approaches to deal with the idiosyncrasies of decision support. We had been groomed in traditional development methodologies, but we modified and enhanced those practices to address the unique challenges of providing data access and analytics to business users, while considering growth and extensibility for the long haul.

Over the years, the authors have been involved with literally hundreds of DW/BI projects in a variety of capacities, including vendor, consultant, IT project team member, and business user. Many of these projects have been wildly successful, some have merely met expectations, and a few have failed in spectacular ways. Each project taught us a lesson. In addition, we have all had the opportunity to learn from many talented individuals and organizations over the years. Our approaches and techniques have been refined over time — and distilled into *The Data Warehouse Lifecycle Toolkit*.

When we first published this book in 1998, we struggled with the appropriate name for our methodology. Someone suggested calling it the Kimball Lifecycle, but Ralph modestly resisted because he felt that many others, in addition to him, contributed to the overall approach.

We eventually determined that the official name would be the Business Dimensional Lifecycle because this moniker reinforced the unique core tenets of our methods. We felt very strongly that successful data warehousing depends on three fundamental concepts:

- Focus on the business.
- Dimensionally structure the data that's delivered to the business via ad hoc queries or reports.
- Iteratively develop the overall data warehouse environment in manageable lifecycle increments rather than attempting a galactic Big Bang.

Rewinding back to the 1990s, we were one of the few organizations emphasizing these core principles at the time, so the Business Dimensional Lifecycle name also differentiated our methods from others in the marketplace. Fast forwarding to today, we still firmly believe in these core concepts; however the industry has evolved since the first edition of the *Lifecycle Toolkit* was published. Now nearly everyone else touts these same principles; they've become mainstream best practices. Vocabulary from our approach including dimension tables, fact tables, and slowly changing dimensions have been embedded in the interfaces of many DW/BI tools. While it's both thrilling and affirming that the concepts have been woven into the fiber of our industry, they're no longer differentiators of our approach. Second, despite our thoughtful naming of the Business Dimensional Lifecycle, the result was a mouthful, so most people in the industry simply refer to our methods as the Kimball approach, anyhow. Therefore, we're officially adopting the Kimball Lifecycle nomenclature going forward.

In spite of dramatic advancements in technology and understanding during the last couple of decades, the basic constructs of the Kimball Lifecycle have remained strikingly constant. Our approach to designing, developing, and deploying DW/BI solutions is tried and true. It has been tested with projects across virtually every industry, business function, and platform. The Kimball Lifecycle approach has proven to work again and again. In fact, that's the reasoning behind the Kimball Group's ''practical techniques, proven results'' motto.

Lifecycle Milestones

The overall Kimball Lifecycle approach to DW/BI initiatives is illustrated in Figure 1-1. Successful implementation of a DW/BI system depends on the appropriate integration of numerous tasks and components. It is not enough to have the perfect data model or best-of-breed technology. You need to coordinate the many facets of a DW/BI project, much like a conductor must unify the many instruments in an orchestra. A soloist cannot carry a full orchestra. Likewise, the DW/BI implementation effort needs to demonstrate strength across all aspects of the project for success. The Kimball Lifecycle is

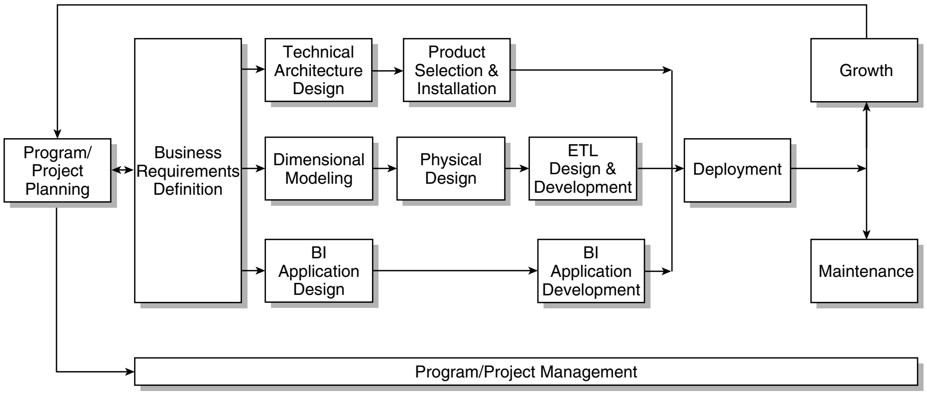
Program/ Project Planning
Business Requirements Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
ETL Design & Development
BI Application Design
BI Application Development
Deployment
Growth
Maintenance
Program/Project Management

Figure 1-1 The Kimball Lifecycle diagram.

similar to the conductor's score. It ensures that the project pieces are brought together in the right order and at the right time.

The Lifecycle diagram depicts the sequence of high level tasks required for effective DW/BI design, development, and deployment. The diagram shows the overall roadmap, while each box serves as a guidepost or mile/kilometer marker. We'll briefly describe the milestones, as well as provide references to the corresponding chapters in this book for more specific driving instructions.

Program/Project Planning

The Lifecycle begins with program and project planning, as one would expect. Throughout this book, *project* refers to a single iteration of the Kimball Lifecycle from launch through deployment; projects have a finite start and end. On the other hand, *program* refers to the broader, ongoing coordination of resources, infrastructure, timelines, and communication across multiple projects; a program is an overall umbrella encompassing more than one project. It should continuously renew itself and should rarely have an abrupt end.

Which comes first, the program or the project? Much like the classic chicken and egg conundrum, it's not always obvious which comes first. In some organizations, executive agreement is reached to launch a DW/BI program and then it's a matter of prioritizing to identify the initial project. In other situations, funding is provided for a single project or two, and then the need for program coordination is subsequently realized. There's no single right approach or sequence.

There's much greater consistency around project planning, beginning with the scoping of the DW/BI project. Obviously, you must have a basic understanding of the business's requirements to make appropriate scope decisions; the bi-directional arrow between the project planning and business requirements boxes in Figure 1-1 shows this dependency. Project planning then turns to resource staffing, coupled with project task identification, assignment, duration, and sequencing. The resulting integrated project plan identifies all tasks associated with the Kimball Lifecycle and the responsible parties. It serves as the cornerstone for the ongoing management of your DW/BI project. Chapter 2 details these launch activities, in addition to the ongoing management of the program/project.

Program/Project Management

Program/project management ensures that the Kimball Lifecycle activities remain on track and in sync. Program/project management activities focus on monitoring project status, issue tracking, and change control to preserve scope boundaries. Ongoing management also includes the development of a comprehensive communication plan that addresses both the business and information technology (IT) constituencies. Continuing communication is critical to managing expectations; managing expectations is critical to achieving your DW/BI goals.

Business Requirements Definition

A DW/BI initiative's likelihood of success is greatly increased by a sound understanding of the business users and their requirements. Without this understanding, DW/BI often becomes a technical exercise in futility for the project team.

Our approach for gathering knowledge workers' analytic requirements differs significantly from more traditional, data-driven requirements analysis. DW/BI analysts must understand the key factors driving the business in order to successfully translate the business requirements into design considerations. An effective business requirements definition is crucial as it establishes the foundation for all downstream Lifecycle activities. Chapter 3 provides a comprehensive discussion of tips and techniques for gathering business requirements.

Technology Track

Following the business requirements definition, there are three concurrent tracks focusing on technology, data, and business intelligence applications, respectively. While the arrows in the Figure 1-1 Lifecycle diagram designate the activity workflow along each of the parallel tracks, there are also implied dependencies between the tasks, as illustrated by the vertical alignment of the task boxes.

The technology track is covered in Chapters 4 and 5. Chapter 4 introduces overall technical architecture concepts, and Chapter 5 focuses on the process of designing your architecture and then selecting products to instantiate it. You can think of these two companion chapters as delivering the ''what,'' followed by the ''how.''

Technical Architecture Design

DW/BI environments require the integration of numerous technologies. The technical architecture design establishes the overall architectural framework and vision. Three factors — the business requirements, current technical environment, and planned strategic technical directions — must be considered simultaneously to establish the appropriate DW/BI technical architecture design. You should resist the natural tendency to begin by focusing on technology in isolation.

Product Selection and Installation

Using your technical architecture plan as a virtual shopping list of needed capabilities, specific architectural components such as the hardware platform, database management system, extract-transformation-load (ETL) tool, or data access query and reporting tool must be evaluated and selected. Once the products have been selected, they are then installed and tested to ensure appropriate end-to-end integration within your DW/BI environment.

Data Track

The second parallel set of activities following the business requirements definition is the data track, from the design of the target dimensional model, to the physical instantiation of the model, and finally the ‘‘heavy lifting’’ where source data is extracted, transformed, and loaded into the target models.

Dimensional Modeling

During the gathering of business requirements, the organization’s data needs are determined and documented in a preliminary *enterprise data warehouse bus matrix* representing the organization’s key business processes and their associated dimensionality. This matrix serves as a data architecture blueprint to ensure that the DW/BI data can be integrated and extended across the organization over time.

Designing dimensional models to support the business’s reporting and analytic needs requires a different approach than that used for transaction processing design. Following a more detailed data analysis of a single business process matrix row, modelers identify the fact table granularity, associated dimensions and attributes, and numeric facts.

These dimensional modeling concepts are discussed in Chapters 6 and 7. Similar to our handling of the technology track, Chapter 6 introduces dimensional modeling concepts, and Chapter 7 describes the recommended approach and process for developing a dimensional model.

Physical Design

Physical database design focuses on defining the physical structures, including setting up the database environment and instituting appropriate security. Although the physical data model in the relational database will be virtually identical to the dimensional model, there are additional issues to address, such as preliminary performance tuning strategies, from indexing to partitioning and aggregations. If appropriate, OLAP databases are also designed during this process. Physical design topics are discussed in Chapter 8.

ETL Design and Development

Design and development of the extract, transformation, and load (ETL) system remains one of the most vexing challenges confronted by a DW/BI project team; even when all the other tasks have been well planned and executed, 70% of the risk and effort in the DW/BI project comes from this step.

Chapter 9 discusses the overall architecture of the ETL system and provides a comprehensive review of the 34 subsystem building blocks that are needed in nearly every data warehouse back room to provide extraction, cleansing and conforming, and delivery and management capabilities. Chapter 10 then converts the subsystem discussion into reality with specific details of the ETL design and development process and associated tasks, including both historical data loads and incremental processing and automation.

Business Intelligence Application Track

The final concurrent activity track focuses on the business intelligence (BI) applications. General concepts and rationale are presented in Chapter 11, and design and development best practices are covered in Chapter 12.

BI Application Design

Immediately following the business requirements definition, while some DW/BI team members are working on the technical architecture and dimensional models, others should be working with the business to identify the candidate BI applications, along with appropriate navigation interfaces to address the users' needs and capabilities. For most business users, parameter-driven BI applications are as ad hoc as they want or need. BI applications are the vehicle for delivering business value from the DW/BI solution, rather than just delivering the data.

BI Application Development

Following BI application specification, application development tasks include configuring the business metadata and tool infrastructure, and then constructing and validating the specified analytic and operational BI applications, along with the navigational portal.

Deployment

The three parallel tracks, focused on technology, data, and BI applications, converge at deployment. Extensive planning is required to ensure that these puzzle pieces are tested and fit together properly, in conjunction with the appropriate education and support infrastructure. As emphasized in Chapter 13, it is critical that deployment be well orchestrated; deployment should be deferred if all the pieces, such as training, documentation, and validated data, are not ready for prime time release.

Maintenance

Once the DW/BI system is in production, technical operational tasks are necessary to keep the system performing optimally, including usage monitoring, performance tuning, index maintenance, and system backup. You must also continue focusing on the business users with ongoing support, education, and communication. These maintenance issues and associated tasks are discussed in Chapter 13.

Growth

If you have done your job well, the DW/BI system is bound to expand and evolve to deliver more value to the business. Unlike traditional systems development initiatives, change should be viewed as a sign of success, not failure. Prioritization processes must be established to deal with the ongoing business demand. We then go back to the beginning of the Lifecycle, leveraging and building upon the foundation that has already been established, while turning our attention to the new requirements. Chapter 14 details recommendations to address the long-term health and growth of your DW/BI environment.

Using the Lifecycle Roadmap

The Kimball Lifecycle diagram in Figure 1-1 illustrates the general flow of a DW/BI implementation. It identifies task sequencing and highlights the activities that should happen concurrently throughout the technology, data, and BI application tracks.

The Lifecycle diagram, however, does not attempt to reflect an absolute project timeline. Each box in Figure 1-1 is the same width, with the exception of program/project management. If you have any experience with data warehousing and business intelligence, you know that the resources and time required for each Lifecycle box are *not* equal. Clearly, the reader should not lay a ruler along the bottom of the diagram and divide the tasks into timeline months; focus on sequencing and concurrency, not absolute timelines.

As with most approaches, you may need to customize the Kimball Lifecycle to address the unique needs of your organization. If this is the case, we applaud your adoption of the framework, as well as your creativity. Truth be told, we usually tailor the specific Lifecycle tasks for each new project. Throughout this book, we attempt to describe nearly everything you need to think about during the design, development, and deployment of a DW/BI solution. Don't let the volume of material overwhelm you. Not every detail of every Lifecycle task will be performed on every project.

Finally, as we'll further describe in Chapter 2, the Kimball Lifecycle is most effective when used to implement projects of manageable, yet meaningful scope. It is nearly impossible to tackle everything at once, so don't let your business users, fellow team members, or management force that approach.

Lifecycle Navigation Aids

Not surprisingly, the book is riddled with references to the Kimball Lifecycle. For starters, each chapter title page includes a miniature graphic of the Lifecycle diagram, highlighting where you are within the overall framework. You should view this as your Lifecycle mile marker. Be forewarned that there is not always a one-to-one relationship between mile markers and book chapters. In some cases, a single chapter addresses multiple markers, as in Chapter 2, which covers both program/project planning and management. In other cases, multiple chapters cover a single mile marker, such as Chapters 6 and 7, which discuss dimensional modeling, or Chapters 9 and 10, which provide detailed coverage of ETL design and development.

In addition to the ''you are here'' mile markers, there's a ''blueprint for action'' at the end of each process-oriented chapter that includes the following guidance and recommendations:

- Managing the effort and reducing risk.
- Assuring quality.
- Key project team roles involved in the process.
- Key deliverables.
- Estimating guidelines.
- Templates and other resources available on the companion book website at `www.kimballgroup.com`.
- Detailed listing of project tasks.

Lifecycle Vocabulary Primer

You are inevitably anxious to jump into the details and move ahead with your DW/BI program/project, but we first want to define several terms that are used throughout this book. We'll also note core vocabulary changes since the first edition of this publication.

Unfortunately, the DW/BI industry is plagued with terminology that's used imprecisely or in contradictory ways. Some of the long-standing debates in our industry are fueled as much from misunderstandings about what others mean by a term, as from true differences in philosophy. Though we can't settle the debates in this forum, we will try to be clear and consistent throughout this text.

Data Warehouse versus Business Intelligence

As an industry, we can't seem to reach consensus about what to call ourselves. Traditionally, the Kimball Group has referred to the overall process of providing information to support business decision making as *data warehousing*. Delivering the entire end-to-end solution, from the source extracts to the queries and applications that the business users interact with, has always been one of our fundamental principles; we would never consider building data warehouse databases without delivering the presentation and access capabilities. This terminology is strongly tied to our legacy of books, articles, and design tips. In fact, nearly all our *Toolkit* books include references to the data warehouse in their titles.

The term *business intelligence* initially emerged in the 1990s to refer to the reporting and analysis of data stored in the warehouse. When it first appeared on the industry's radar, several of this book's authors were dumbfounded about the hoopla it was generating because we'd been advocating the practices for years. It wasn't until we dug a little deeper that we discovered many organizations had built data warehouses as if they were archival librarians, without any regard to getting the data out and delivered to the business users in a useful manner. No wonder earlier data warehouses had failed and people were excited about BI as a vehicle to deliver on the promise of business value!

Some folks in our industry continue to refer to data warehousing as the overall umbrella term, with the data warehouse databases and BI layers as subset deliverables within that context. Alternatively, others refer to business intelligence as the overarching term, with the data warehouse relegated to describe the central data store foundation of the overall business intelligence environment.

Because the industry has not reached agreement, we consistently use the phrase ''data warehouse/business intelligence'' (DW/BI) to mean the complete end-to-end system. Though some would argue that you can theoretically deliver BI without a data warehouse, and vice versa, that is ill-advised from our perspective. Linking the two together in the DW/BI acronym further reinforces their dependency.

Independently, we refer to the queryable data in your DW/BI system as the *enterprise data warehouse*, and value-add analytics as *BI applications*. In other words, the *data warehouse is the foundation for business intelligence*. We disagree with others who insist that the data warehouse is a highly normalized data store whose primary purpose is not query support, but to serve as a source for the transformation and loading of data into summarized dimensional structures.

ETL System

We often refer to the extract, transformation, and load (ETL) system as the back room kitchen of the DW/BI environment. In a commercial restaurant's kitchen, raw materials are dropped off at the back door and then transformed into a delectable meal for the restaurant patrons by talented chefs. Long before a commercial kitchen is put into productive use, a significant amount of planning goes into the workspace and components' blueprint.

The restaurant's kitchen is designed for efficiency, while at the same time ensuring high quality and integrity. Kitchen throughput is critical when the restaurant is packed with patrons, but the establishment is doomed if the meals coming out of the kitchen are inconsistent, fail to meet expectations, or worse, cause food poisoning. Chefs strive to procure high quality products and reject those that don't meet their standards.

Skilled kitchen professionals wield the tools of their trade. Due to the sharp knives and hot surfaces in the kitchen, restaurant patrons aren't invited behind the scenes to check out the food preparation or taste the sauce before ordering an entree. It's just not safe, plus there's a variety of ''processing'' in the kitchen that patrons just shouldn't be privy to.

Much the same holds true for the DW/BI kitchen. Raw data is extracted from the operational source systems and dumped into the kitchen where it is transformed into meaningful information for the business. The ETL area must be laid out and architected long before any data is extracted from the source. The ETL system strives to deliver high throughput, as well as high quality output. Incoming data is checked for reasonable quality; data quality conditions are continuously monitored.

Skilled ETL architects and developers wield the tools of their trade in the DW/BI kitchen; business users and BI applications are barred from entering the ETL system and querying the associated work-in-process files before the data is quality assured and ready for business consumption. ETL professionals and the system throughput shouldn't be compromised by unpredictable inquiries. Once the data is verified and ready for business consumption, it is appropriately arranged ''on the plate'' and brought through the door into the DW/BI front room.

In this edition of *Lifecycle Toolkit*, we have greatly expanded our coverage of ETL architecture best practices, largely because we observed so many DW/BI teams taking a haphazard approach to designing and developing their kitchen.

The introduction of 34 subsystems provides a formidable checklist for anyone constructing or remodeling an ETL kitchen.

For readers who are familiar with the first edition, we have abandoned the *data staging* terminology due to several developments. When the book was originally written, ETL had not been established as an industry standard acronym. And while we consistently used data staging to refer to all the cleansing and data preparation processing that occurred between the source extraction and loading into target databases, others used the term to merely mean the initial dumping of raw source data into a work zone.

Business Process Dimensional Model

Now let's turn our attention to the restaurant's dining room where the focus shifts to the patrons' overall dining experience. Patrons want quality food, appealing décor, prompt service, and reasonable cost. The dining room is designed and managed based on the preferences expressed by the restaurant's patrons, not the kitchen staff.

Similarly, the DW/BI system's front room must be designed and managed with the business users' needs first and foremost at all times. Dimensional models are a fundamental front room deliverable. Dimensional modeling is a design discipline optimized to deliver on the twin goals of business users' ease of use and BI query performance. Dimensional models contain the same data content and relationships as models normalized into third normal form; they're just structured differently. Normalized models are optimized for high volume, single row inserts and updates as typified by transaction processing systems, but they fail to deliver the understandability and query performance required by DW/BI.

The two primary constructs of a dimensional model are fact tables and dimension tables. Fact tables contain the metrics resulting from a business process or measurement event, such as the sales ordering process or service call event. While it may appear as a subtlety for the casual reader, our business process orientation has widespread ramifications throughout the Lifecycle. We put a deep stake in the ground about the importance of structuring dimensional models around business processes and their associated data sources, instead of taking a business department/function or analytic reporting perspective advocated by others in the industry. This allows us to design identical, consistent views of data for all observers, regardless of which department they belong to, which goes a long way toward eliminating misunderstandings at business meetings!

We also feel strongly about the need for precise declaration of the fact table's grain at the lowest, most atomic level captured by the business process for maximum flexibility and extensibility. Atomic data lets business users ask constantly changing, free-ranging, and very precise questions. It is unacceptable to have this robust data locked in normalized schemas where it is unable to quickly and easily respond to business queries.

Dimension tables contain the descriptive attributes and characteristics associated with specific, tangible measurement events, such as the customer, product, or sales representative associated with an order being placed. Dimension attributes are used for constraining, grouping, or labeling in a query. Hierarchical many-to-one relationships are denormalized into single dimension tables. *Conformed dimensions* are the master data of the DW/BI environment, managed once in the kitchen and then shared by multiple dimensional models to enable enterprise integration and ensure consistency.

Dimensional models may be physically instantiated in a relational database, in which case they're often referred to as *star schema*. Alternatively, dimensional models can also be stored in an online analytic processing (OLAP) database where they're commonly referred to as *cubes*. We recommend that the OLAP cubes be populated from the relational atomic dimensional models for operational reasons.

In the first edition of *Lifecycle Toolkit*, we used the term *data mart* extensively instead of *business process dimensional models*. While data mart wins the brevity competition, the term has been marginalized by others to mean summarized departmental, independent non-architected datasets.

Business Intelligence Applications

It's not enough to just deliver dimensional data to the DW/BI system's front dining room. While some business users are interested in and capable of formulating ad hoc queries on the spur of the moment, the majority of the business community will be more satisfied with the ability to execute predefined applications that query, analyze, and present information from the dimensional model. There is a broad spectrum of *BI application* capabilities, ranging in complexity from a set of canned static reports to analytic applications that directly interact with the operational transaction systems. In all cases, the goal is to deliver capabilities that are accepted by the business to support and enhance their decision making. Clearly, the BI applications in your dining room must address the patron's needs, be organized to their liking, and deliver results in an acceptable timeframe. While you're at it, you'll also want to provide a ''menu'' to describe what's available; fortunately, metadata should come to the rescue on this front.

There's one final nomenclature change that we want to make you aware of. In the first edition, we referred to these templates and applications as *end user applications*, instead of the more current BI application terminology.

Conclusion

The Kimball Lifecycle provides the framework to organize the numerous tasks required to implement a successful DW/BI system. It has evolved through years of hands-on experience and is firmly grounded in the realities you face today. Now with the Lifecycle framework in mind, let's get started!