

17

Kimball DW/BI Lifecycle Overview

The gears shift rather dramatically in this chapter. Rather than focusing on Kimball dimensional modeling techniques, we turn your attention to everything else that occurs during the course of a data warehouse/business intelligence design and implementation project. In this chapter, we'll cover the life of a DW/BI project from inception through ongoing maintenance, identifying best practices at each step, as well as potential vulnerabilities. More comprehensive coverage of the Kimball Lifecycle is available in *The Data Warehouse Lifecycle Toolkit, Second Edition* by Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, and Bob Becker (Wiley, 2008). This chapter is a crash course drawn from the complete text, which weighs in at a hefty 600+ pages.

You may perceive this chapter's content is only applicable to DW/BI project managers, but we feel differently. Implementing a DW/BI system requires tightly integrated activities. We believe everyone on the project team, including the analysts, architects, designers, and developers, needs a high-level understanding of the complete Lifecycle.

This chapter provides an overview of the entire Kimball Lifecycle approach; specific recommendations regarding dimensional modeling and ETL tasks are deferred until subsequent chapters. We will dive into the collaborative modeling workshop process in Chapter 18: Dimensional Modeling Process and Tasks, then make a similar plunge into ETL activities in Chapter 20: ETL System Design and Development Process and Tasks.

Chapter 17 covers the following concepts:

- Kimball Lifecycle orientation
- DW/BI program/project planning and ongoing management
- Tactics for collecting business requirements, including prioritization
- Process for developing the technical architecture and selecting products

- Physical design considerations, including aggregation and indexing
- BI application design and development activities
- Recommendations for deployment, ongoing maintenance, and future growth

Lifecycle Roadmap

When driving to a place we've never been to before, most of us rely on a roadmap, albeit displayed via a GPS. Similarly, a roadmap is extremely useful if we're about to embark on the unfamiliar journey of data warehousing and business intelligence. The authors of *The Data Warehouse Lifecycle Toolkit* drew on decades of experience to develop the Kimball Lifecycle approach. When we first introduced the Lifecycle in 1998, we referred to it as the Business Dimensional Lifecycle, a name that reinforced our key tenets for data warehouse success: Focus on the business's needs, present dimensionally structured data to users, and tackle manageable, iterative projects. In the 1990s, we were one of the few organizations emphasizing these core principles, so the moniker differentiated our methods from others. We are still very firmly wed to these principles, which have since become generally-accepted industry best practices, but we renamed our approach to be the Kimball Lifecycle because that's how most people refer to it.

The overall Kimball Lifecycle approach is encapsulated in Figure 17-1. The diagram illustrates task sequence, dependency, and concurrency. It serves as a roadmap to help teams do the right thing at the right time. The diagram does not reflect an absolute timeline; although the boxes are equally wide, there's a vast difference in the time and effort required for each major activity.

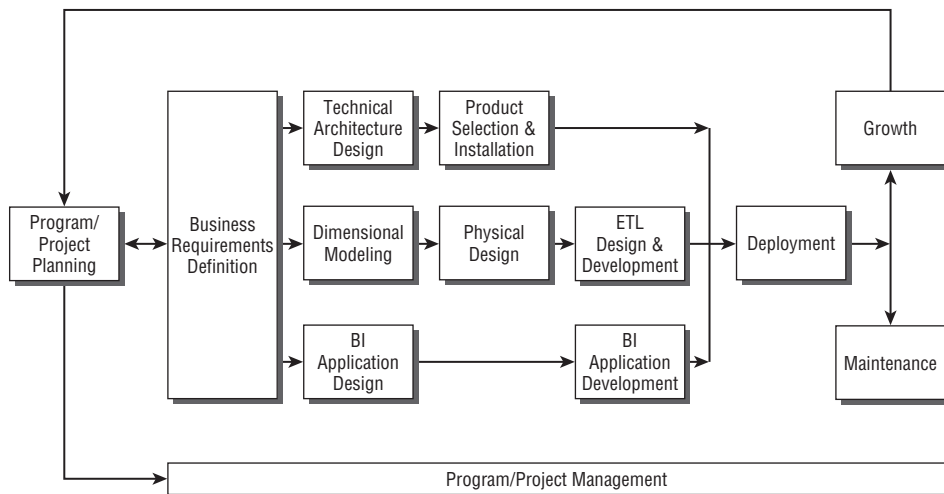


Figure 17-1: Kimball Lifecycle diagram.

NOTE Given the recent industry focus on agile methodologies, we want to remind readers about the discussion of the topic in Chapter 1: Data Warehousing, Business Intelligence, and Dimensional Modeling Primer. The Kimball Lifecycle approach and agile methodologies share some common doctrines: Focus on business value, collaborate with the business, and develop incrementally. However, we also feel strongly that DW/BI system design and development needs to be built on a solid data architecture and governance foundation, driven by the bus architecture. We also believe most situations warrant the bundling of multiple agile “deliverables” into a more full-function release before being broadly deployed to the general business community.

Roadmap Mile Markers

Before diving into specifics, take a moment to orient yourself to the roadmap. The Lifecycle begins with program/project planning, as you would expect. This module assesses the organization’s readiness for a DW/BI initiative, establishes the preliminary scope and justification, obtains resources, and launches the program/project. Ongoing project management serves as a foundation to keep the remaining activities on track.

The second major task in Figure 17-1 focuses on business requirements definition. There’s a two-way arrow between program/project planning and the business requirements definition due to the interplay between these activities. Aligning the DW/BI initiative with business requirements is absolutely crucial. Best-of-breed technologies won’t salvage a DW/BI environment that fails to focus on the business. Business users and their requirements have an impact on almost every design and implementation decision made during the course of a DW/BI project. In Figure 17-1’s roadmap, this is reflected by the three parallel tracks that follow.

The top track of Figure 17-1 deals with technology. Technical architecture design establishes the overall framework to support the integration of multiple technologies. Using the capabilities identified in the architecture design as a shopping list, you then evaluate and select specific products. Notice that product selection is not the first box on the roadmap. One of the most frequent mistakes made by novice teams is to select products without a clear understanding of what they’re trying to accomplish. This is akin to grabbing a hammer whether you need to pound a nail or tighten a screw.

The middle track emanating from business requirements definition focuses on data. It begins by translating the requirements into a dimensional model, as we’ve been practicing. The dimensional model is then transformed into a physical structure. The focus is on performance tuning strategies, such as aggregation, indexing, and partitioning, during the physical design. Last but not least, the ETL system is

designed and developed. As mentioned earlier, the equally sized boxes don't represent equally sized efforts; this becomes obvious with the workload differential between the physical design and the demanding ETL-centric activities.

The final set of tasks spawned by the business requirements is the design and development of the BI applications. The DW/BI project isn't done when you deliver data. BI applications, in the form of parameter-driven templates and analyses, will satisfy a large percentage of the business users' analytic needs.

The technology, data, and BI application tracks, along with a healthy dose of education and support, converge for a well-orchestrated deployment. From there, on-going maintenance is needed to ensure the DW/BI system remains healthy. Finally, you handle future growth by initiating subsequent projects, each returning to the beginning of the Lifecycle all over again.

Now that you have a high-level understanding of the overall roadmap, we'll describe each of the boxes in Figure 17-1 in more detail.

Lifecycle Launch Activities

The following sections outline best practices, and pitfalls to avoid, as you launch a DW/BI project.

Program/Project Planning and Management

Not surprisingly, the DW/BI initiative begins with a series of program and project planning activities.

Assessing Readiness

Before moving ahead with a DW/BI effort, it is prudent to take a moment to assess the organization's readiness to proceed. Based on our cumulative experience from hundreds of client engagements, three factors differentiate projects that were predominantly smooth sailing versus those that entailed a constant struggle. These factors are leading indicators of DW/BI success; we'll describe the characteristics in rank order of importance.

The most critical readiness factor is to have a strong executive business sponsor. Business sponsors should have a clear vision for the DW/BI system's potential impact on the organization. Optimally, business sponsors have a track record of success with other internal initiatives. They should be politically astute leaders who can convince their peers to support the effort. It's a much riskier scenario if the chief information officer (CIO) is the designated sponsor; we much prefer visible commitment from a business partner-in-crime instead.

The second readiness factor is having a strong, compelling business motivation for tackling the DW/BI initiative. This factor often goes hand in hand with sponsorship. The DW/BI project needs to solve critical business problems to garner the resources required for a successful launch and healthy lifespan. Compelling motivation typically creates a sense of urgency, whether the motivation is from external sources, such as competitive factors, or internal sources, such as the inability to analyze cross-organization performance following acquisitions.

The third factor when assessing readiness is feasibility. There are several aspects of feasibility, including technical and resource feasibility, but data feasibility is the most crucial. Are you collecting real data in real operational source systems to support the business requirements? Data feasibility is a major concern because there is no short-term fix if you're not already collecting reasonably clean source data at the right granularity.

Scoping and Justification

When you're comfortable with the organization's readiness, it's time to put boundaries around an initial project. Scoping requires the joint input of the IT organization and business management. The scope of a DW/BI project should be both meaningful to the business organization and manageable for the IT organization. You should initially tackle projects that focus on data from a single business process; save the more challenging, cross-process projects for a later phase. Remember to avoid the *Law of Too* when scoping—too brief of a timeline for a project with too many source systems and too many users in too many locations with too diverse analytic requirements.

Justification requires an estimation of the benefits and costs associated with the DW/BI initiative. Hopefully, the anticipated benefits grossly outweigh the costs. IT usually is responsible for deriving the expenses. DW/BI systems tend to expand rapidly, so be sure the estimates allow room for short-term growth. Unlike operational system development where resource requirements tail off after production, ongoing DW/BI support needs will not decline appreciably over time.

The business community should have prime responsibility for determining the anticipated financial benefits. DW/BI environments typically are justified based on increased revenue or profit opportunities rather than merely focusing on expense reduction. Delivering “a single version of the truth” or “flexible access to information” isn't sufficient financial justification. You need to peel back the layers to determine the quantifiable impact of improved decision making made possible by these sound bites. If you are struggling with justification, this is likely a symptom that the initiative is focused on the wrong business sponsor or problem.

Staffing

DW/BI projects require the integration of a cross-functional team with resources from both the business and IT communities. It is common for the same person to fill multiple roles on the team; the assignment of named resources to roles depends on the project's magnitude and scope, as well as the individual's availability, capacity, and experience.

From the business side of the house, we'll need representatives to fill the following roles:

- **Business sponsor.** The sponsor is the DW/BI system's ultimate client, as well as its strongest advocate. Sponsorship sometimes takes the form of an executive steering committee, especially for cross-enterprise initiatives.
- **Business driver.** In a large organization, the sponsor may be too far removed or inaccessible to the project team. In this case, the sponsor sometimes delegates less strategic DW/BI responsibilities to a middle manager in the organization. This driver should possess the same characteristics as the sponsor.
- **Business lead.** The business project lead is a well-respected person who is highly involved in the project, communicating with the project manager on a daily basis. Sometimes the business driver fills this role.
- **Business users.** Optimally, the business users are the enthusiastic fans of the DW/BI environment. You need to involve them early and often, beginning with the project scope and business requirements. From there, you must find creative ways to maintain their interest and involvement throughout the project. Remember, business user involvement is critical to DW/BI acceptance. Without business users, the DW/BI system is a technical exercise in futility.

Several positions are staffed from either the business or IT organizations. These straddlers can be technical resources who understand the business or business resources who understand technology:

- **Business analyst.** This person is responsible for determining the business needs and translating them into architectural, data, and BI application requirements.
- **Data steward.** This subject matter expert is often the current go-to resource for ad hoc analysis. They understand what the data means, how it is used, and where data inconsistencies are lurking. Given the need for organizational consensus around core dimensional data, this can be a politically challenging role, as we described in Chapter 4: Inventory.
- **BI application designer/developer.** BI application resources are responsible for designing and developing the starter set of analytic templates, as well as providing ongoing BI application support.

The following roles are typically staffed from the IT organization:

- **Project manager.** The project manager is a critical position. This person should be comfortable with and respected by business executives, as well as technical resources. The project manager's communication and project management skills must be stellar.
- **Technical architect.** The architect is responsible for the overall technical architecture. This person develops the plan that ties together the required technical functionality and helps evaluate products on the basis of the overall architecture.
- **Data architect/modeler.** This resource likely comes from a transactional data background with heavy emphasis on normalization. This person should embrace dimensional modeling concepts and be empathetic to the requirements of the business rather than focused strictly on saving space or reducing the ETL workload.
- **Database administrator.** Like the data modeler, the database administrator must be willing to set aside some traditional database administration truisms, such as having only one index on a relational table.
- **Metadata coordinator.** This person helps establish the metadata repository strategy and ensures that the appropriate metadata is collected, managed, and disseminated.
- **ETL architect/designer.** This role is responsible for designing the ETL environment and processes.
- **ETL developer.** Based on direction from the ETL architect/designer, the developer builds and automates the processes, likely using an ETL tool.

We want to point out again that this is a list of roles, not people. Especially in smaller shops, talented individuals will fill many of these roles simultaneously.

Developing and Maintaining the Plan

The DW/BI project plan identifies all the necessary Lifecycle tasks. A detailed task list is available on the Kimball Group website at www.kimballgroup.com; check out the Tools & Utilities tab under *The Data Warehouse Lifecycle Toolkit, Second Edition* book title.

Any good project manager knows key team members should develop estimates of the effort required for their tasks; the project manager can't dictate the amount of time allowed and expect conformance. The project plan should identify acceptance checkpoints with business representatives after every major roadmap milestone and deliverable to ensure the project remains on track.

DW/BI projects demand broad communication. Although project managers typically excel at intra-team communications, they should also establish

a communication strategy describing the frequency, forum, and key messaging for other constituencies, including the business sponsors, business community, and other IT colleagues.

Finally, DW/BI projects are vulnerable to scope creep largely due to a strong need to satisfy business users' requirements. You have several options when confronted with changes: Increase the scope (by adding time, resources, or budget), play the zero-sum game (by retaining the original scope by giving up something in exchange), or say "no" (without actually saying "no" by handling the change as an enhancement request). The most important thing about scope decisions is that they shouldn't be made in an IT vacuum. The right answer depends on the situation. Now is the time to leverage the partnership with the business to arrive at an answer that everyone can live with.

Business Requirements Definition

Collaborating with business users to understand their requirements and ensure their buy-in is absolutely essential to successful data warehousing and business intelligence. This section focuses on back-to-basics techniques for gathering business requirements.

Requirements Preplanning

Before sitting down with business representatives to collect their requirements, we suggest the following to ensure productive sessions:

Choose the Forum

Business user requirements sessions are typically interwoven with source system expert data discovery sessions. This dual-pronged approach gives you insight into the needs of the business with the realities of the data. However, you don't ask business representatives about the granularity or dimensionality of their critical data. You need to talk to them about what they do, why they do it, how they make decisions, and how they hope to make decisions in the future. Like organizational therapy, you're trying to detect the issues and opportunities.

There are two primary techniques for gathering requirements: interviews or facilitated sessions. Both have their advantages and disadvantages. Interviews encourage individual participation and are also easier to schedule. Facilitated sessions may reduce the elapsed time to gather requirements but require more time commitment from each participant.

Based on our experience, surveys are not a reasonable tool for gathering requirements because they are flat and two-dimensional. The self-selected respondents answer only the questions we've thought to ask in advance; there's no option to probe more deeply. In addition, survey instruments do not help forge the bond between business users and the DW/BI initiative that we strive for.

We generally use a hybrid approach with interviews to gather the details and then facilitation to bring the group to consensus. Although we'll describe this hybrid approach in more detail, much of the discussion applies to pure facilitation as well. The requirements gathering forum choice depends on the team's skills, the organization's culture, and what the business users have already been subjected to. One size definitely does not fit all.

Identify and Prepare the Requirements Team

Regardless of the approach, you need to identify and prepare the involved project team members. If you're doing interviews, you need to identify a lead interviewer whose primary responsibility is to ask great open-ended questions. Meanwhile, the interview scribe takes copious notes. Although a tape recorder may provide more complete coverage of each interview, we don't use one because it changes the meeting dynamics. Our preference is to have a second person in the room with another brain and a set of eyes and ears rather than relying on technology. We often invite one or two additional project members (depending on the number of interviewees) as observers, so they can hear the users' input directly.

Before sitting down with business users, you need to make sure you're approaching the sessions with the right mindset. Don't presume you already know it all; you will definitely learn more about the business during the sessions. On the other hand, you should do some homework by researching available sources, such as the annual report, website, and internal organization chart.

Because the key to getting the right answers is asking the right questions, we recommend drafting questionnaires. The questionnaire should not be viewed as a script; it is a tool to organize your thoughts and serve as a fallback device in case your mind goes blank during the session. The questionnaire will be updated throughout the interview process as the team becomes better versed in the business's subject matter.

Select, Schedule, and Prepare Business Representatives

If this is your first foray into DW/BI, or an effort to develop a cohesive strategy for dealing with existing data stovepipes, you should talk to business people representing a reasonable horizontal spectrum of the organization. This coverage is critical to formulating the enterprise data warehouse bus matrix blueprint. You need to understand the common data and vocabulary across core business functions to build an extensible environment.

Within the target user community, you should cover the organization vertically. DW/BI project teams naturally gravitate toward the business's power analysts. Although their insight is valuable, you can't ignore senior executives and middle management. Otherwise, you are vulnerable to being overly focused on the tactical here-and-now and lose sight of the group's strategic direction.

Scheduling the business representatives can be the most onerous requirements task; be especially nice to the department's administrative assistants. We prefer to meet with executives individually. Meeting with a homogeneous group of two to three people is appropriate for those lower on the organization chart. Allow 1 hour for individual meetings and 1½ hours for small groups. The scheduler needs to allow ½ hour between meetings for debriefing and other necessities. Interviewing is extremely taxing due to the focus required. Consequently, don't schedule more than three to four sessions in a day.

When it comes to preparing the interviewees, the business sponsor should communicate with them, stressing their commitment to the effort and the importance of everyone's participation. The interviewees should be asked to bring copies of their key reports and analyses to the session. This communication disseminates a consistent message about the project, plus conveys the business's ownership of the initiative. Occasionally interviewees will be reluctant to bring the business's "crown jewel" reports to the meeting, especially with an outside consultant. However, almost always we have found these people will enthusiastically race back to their offices at the end of the interview to bring back those same reports.

Collecting Business Requirements

It's time to sit down face-to-face to gather the business's requirements. The process usually flows from an introduction through structured questioning to a final wrap-up.

Launch

Responsibility for introducing the session should be established prior to gathering in a conference room. The designated kickoff person should script the primary talking points for the first few minutes when the tone of the interview meeting is set. The introduction should convey a crisp, business-centric message and not ramble with hardware, software, and other technical jargon.

Interview Flow

The objective of an interview is to get business users to talk about what they do and why they do it. A simple, nonthreatening place to begin is to ask about job responsibilities and organizational fit. This is a lob-ball that interviewees can easily respond to. From there, you typically ask about their key performance metrics. Determining how they track progress and success translates directly into the dimensional model; they're telling you about their key business processes and facts without you asking those questions directly.

If you meet with a person who has more hands-on data experience, you should probe to better understand the business's dimensionality. Questions like "How do you distinguish between products (or agents, providers, or facilities)?" or "How

do you naturally categorize products?” help identify key dimension attributes and hierarchies.

If the interviewee is more analytic, ask about the types of analysis currently generated. Understanding the nature of these analyses and whether they are ad hoc or standardized provides input into the BI tool requirements, as well as the BI application design process. Hopefully, the interviewee brought along copies of key spreadsheets and reports. Rather than stashing them in a folder, it is helpful to understand how the interviewee uses the analysis today, as well as opportunities for improvement. Contrary to the advice of some industry pundits, you cannot design an extensible analytic environment merely by getting users to agree on their top five reports. The users’ questions are bound to change; consequently, you must resist the temptation to narrow your design focus to a supposed top five.

If you meet with business executives, don’t dive into these tactical details. Instead, ask them about their vision for better leveraging information throughout the organization. Perhaps the project team is envisioning a totally ad hoc environment, whereas business management is more interested in the delivery of standardized analyses. You need to ensure the DW/BI deliverable matches the business demand and expectations.

Ask each interviewee about the impact of improved access to information. You likely already received preliminary project funding, but it never hurts to capture more potential, quantifiable benefits.

Wrap-Up

As the interview is coming to a conclusion, ask each interviewee about their success criteria for the project. Of course, each criterion should be measurable. “Easy to use” and “fast” mean something different to everyone, so the interviewees need to articulate specifics, such as their expectations regarding the amount of training required to run predefined BI reports.

At this point, always make a broad disclaimer. The interviewees must understand that just because you discussed a capability in the meeting doesn’t guarantee it’ll be included in the first phase of the project. Thank interviewees for their brilliant insights, and let them know what’s happening next and what their involvement will be.

Conducting Data-Centric Interviews

While we’re focused on understanding the business’s requirements, it is helpful to intersperse sessions with the source system data gurus or subject matter experts to evaluate the feasibility of supporting the business needs. These data-focused interviews are quite different from the ones just described. The goal is to ascertain whether the necessary core data exists before momentum builds behind the requirements. In these data-centric interviews, you may go so far as to ask for some initial data profiling results, such as domain values and counts of a few critical

data fields, to be provided subsequently, just to ensure you are not standing on quicksand. A more complete data audit will occur during the dimensional modeling process. Try to learn enough at this point to manage the organization's expectations appropriately.

Documenting Requirements

Immediately following the interview, the interview team should debrief. You must ensure everyone is on the same page about what was learned. It is also helpful if everyone reviews their notes shortly after the session to fill in gaps while the interview is still fresh. Abbreviations and partial sentences in the notes become incomprehensible after a few days! Likewise, examine the reports gathered to gain further insight into the dimensionality that must be supported in the data warehouse.

At this point, it is time to document what you've heard. Although documentation is everyone's least favorite activity, it is critical for both user validation and project team reference materials. There are two potential levels of documentation resulting from the requirements process. The first is to write up each individual interview; this activity is time-consuming because the write-up should not be merely a stream-of-consciousness transcript but should make sense to someone who wasn't in the interview. The more critical documentation is a consolidated findings document. This document is organized around key business processes. Because you tackle DW/BI projects on a process-by-process basis, it is appropriate to structure the business's requirements into the same buckets that will, in turn, become implementation efforts.

When writing up the findings document, you should begin with an executive summary, followed by a project overview covering the process used and participants involved. The bulk of the document centers on the business processes; for each process, describe why business users want to analyze the process's performance metrics, what capabilities they want, their current limitations, and potential benefits or impact. Commentary about the feasibility of tackling each process is also important.

As described in Chapter 4 and illustrated in Figure 4-11, the processes are sometimes unveiled in an opportunity/stakeholder matrix to convey the impact across the organization. In this case, the rows of the opportunity matrix identify business processes, just like a bus matrix. However, in the opportunity matrix, the columns identify the organizational groups or functions. Surprisingly, this matrix is usually quite dense because many groups want access to the same core performance metrics.

Prioritizing Requirements

The consolidated findings document serves as the basis for presentations back to senior management and other requirements participants. Inevitably you uncovered more than can be tackled in a single iteration, so you need to prioritize. As discussed with project scope, don't make this decision in a vacuum; you need to

leverage (or foster) your partnership with the business community to establish appropriate priorities.

The requirements wrap-up presentation is positioned as a findings review and prioritization meeting. Participants include senior business representatives (who optimally participated in the interviews), as well as the DW/BI manager and other senior IT management. The session begins with an overview of each identified business process. You want everyone in the room to have a common understanding of the opportunities. Also review the opportunity/stakeholder matrix, as well as a simplified bus matrix.

After the findings have been presented, it is time to prioritize using the prioritization grid, illustrated in Figure 17-2. The grid's vertical axis refers to the potential impact or value to the business. The horizontal axis conveys feasibility. Each of the finding's business process themes is placed on the grid based on the representatives' composite agreement regarding impact and feasibility. It's visually obvious where you should begin; projects that warrant immediate attention are located in the upper-right corner because they're high-impact projects, as well as highly feasible. Projects in the lower-left cell should be avoided like the plague; they're missions impossible that do little for the business. Likewise, projects in the lower-right cell don't justify short-term attention, although project teams sometimes gravitate here because these projects are doable but not very crucial. Finally, projects in the upper-left cell represent meaningful opportunities. These projects have large potential business payback but are currently infeasible. While the DW/BI project team focuses on projects in the shaded upper-right corner, other IT teams should address the current feasibility limitations of those in the upper left.

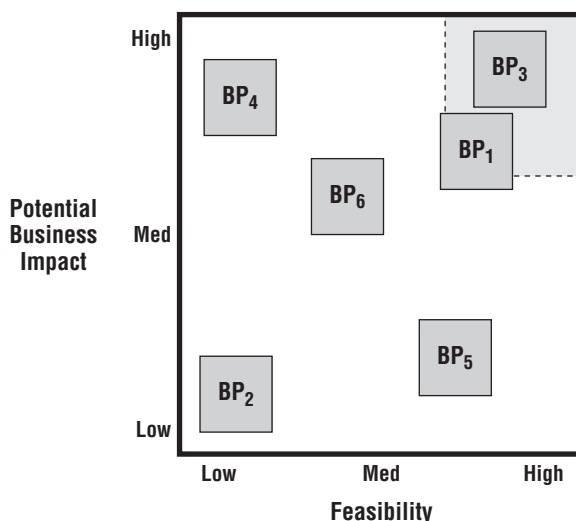


Figure 17-2: Prioritization grid based on business impact and feasibility.

Lifecycle Technology Track

On the Kimball Lifecycle roadmap in Figure 17-1, the business requirements definition is followed immediately by three concurrent tracks focused on technology, data, and BI applications, respectively. In the next several sections we'll zero in on the technology track.

Technical Architecture Design

Much like a blueprint for a new home, the technical architecture is the blueprint for the DW/BI environment's technical services and infrastructure. As the enterprise data warehouse bus architecture introduced in Chapter 4 supports data integration, the architecture plan is an organizing framework to support the integration of technologies and applications.

Like housing blueprints, the technical architecture consists of a series of models that unveil greater detail regarding each major component. In both situations, the architecture enables you to catch problems on paper (such as having the dishwasher too far from the sink) and minimize mid-project surprises. It supports the coordination of parallel efforts while speeding development through the reuse of modular components. The architecture identifies immediately required components versus those that will be incorporated at a later date (such as the deck and screened porch). Most important, the architecture serves as a communication tool. Home construction blueprints enable the architect, general contractor, subcontractors, and homeowner to communicate from a common document. Likewise, the DW/BI technical architecture supports communication regarding a consistent set of technical requirements within the team, upward to management, and outward to vendors.

In Chapter 1, we discussed several major components of the architecture, including ETL and BI services. In this section, we focus on the process of creating the architecture design.

DW/BI teams typically approach the architecture design process from opposite ends of the spectrum. Some teams simply don't understand the benefits of an architecture and feel that the topic and tasks are too nebulous. They're so focused on delivery that the architecture feels like a distraction and impediment to progress, so they opt to bypass architecture design. Instead, they piece together the technical components required for the first iteration with chewing gum and bailing wire, but the integration and interfaces get taxed as more data, more users, or more functionality are added. Eventually, these teams often end up rebuilding because the non-architected structure couldn't withstand the stresses. At the other extreme, some teams want to invest two years designing the architecture while forgetting that the primary purpose of a DW/BI environment is to solve business problems, not address any plausible (and not so plausible) technical challenge.

Neither end of the spectrum is healthy; the most appropriate response lies somewhere in the middle. We've identified the following eight-step process to help navigate these architectural design waters. Every DW/BI system has a technical architecture; the question is whether it is planned and explicit or merely implicit.

Establish an Architecture Task Force

It is useful to create a small task force of two to three people focused on architecture design. Typically, it is the technical architect, along with the ETL architect/designer and BI application architect/designer who ensure both back room and front room representation.

Collect Architecture-Related Requirements

As illustrated in Figure 17-1, defining the technical architecture is not the first box in the Lifecycle diagram. The architecture is created to support business needs; it's not meant to be an excuse to purchase the latest, greatest products. Consequently, key input into the design process should come from the business requirements definition. However, you should listen to the business's requirements with a slightly different filter to drive the architecture design. The primary focus is uncovering the architectural implications associated with the business's needs. Listen closely for timing, availability, and performance requirements.

You should also conduct additional interviews within the IT organization. These are technology-focused sessions to understand current standards, planned technical directions, and nonnegotiable boundaries. In addition, you should uncover lessons learned from prior information delivery projects, as well as the organization's willingness to accommodate operational change on behalf of the DW/BI initiative, such as identifying updated transactions in the source system.

Document Architecture Requirements

After leveraging the business requirements process and conducting supplemental IT interviews, you need to document your findings. We recommend using a simplistic tabular format, just listing each business requirement impacting the architecture, along with a laundry list of architectural implications. For example, if there is a need to deliver global sales performance data on a nightly basis, the technical implications might include 24/7 worldwide availability, data mirroring for loads, robust metadata to support global access, adequate network bandwidth, and sufficient ETL horsepower to handle the complex integration of operational data.

Create the Architecture Model

After the architecture requirements have been documented, you should begin formulating models to support the identified needs. At this point, the architecture team often sequesters itself in a conference room for several days of heavy thinking.

The architecture requirements are grouped into major components, such as ETL, BI, metadata, and infrastructure. From there the team drafts and refines the high-level architectural model. This drawing is similar to the front elevation page on housing blueprints. It illustrates what the architecture will look like from the street, but it can be dangerously simplistic because significant details are embedded in the pages that follow.

Determine Architecture Implementation Phases

Like the homeowner's dream house, you likely can't implement all aspects of the technical architecture at once. Some are nonnegotiable mandatory capabilities, whereas others are nice-to-haves. Again, refer back to the business requirements to establish architecture priorities because you must minimally provide the architectural elements needed to deliver the initial project.

Design and Specify the Subsystems

A large percentage of the needed functionality will likely be met by the major tool vendor's standard offerings, but there are always a few subsystems that may not be found in off-the-shelf products. You must define these subsystems in enough detail, so either someone can build it for you or you can evaluate products against your needs.

Create the Architecture Plan

The technical architecture needs to be documented, including the planned implementation phases, for those who were not sequestered in the conference room. The technical architecture plan document should include adequate detail so skilled professionals can proceed with construction of the framework, much like carpenters frame a house based on the blueprint. However, it doesn't typically reference specific products, except those already in-house.

Review and Finalize the Technical Architecture

Eventually we come full circle with the architecture design process. The architecture task force needs to communicate the architecture plan at varying levels of detail to the project team, IT colleagues, and business leads. Following the review, documentation should be updated and put to use immediately in the product selection process.

Product Selection and Installation

In many ways the architecture plan is similar to a shopping list for selecting products that fit into the plan's framework. The following six tasks associated with DW/BI product selection are quite similar to any technology selection.

Understand the Corporate Purchasing Process

The first step before selecting new products is to understand the internal hardware and software purchase processes.

Develop a Product Evaluation Matrix

Using the architecture plan as a starting point, a spreadsheet-based evaluation matrix should be developed that identifies the evaluation criteria, along with weighting factors to indicate importance; the more specific the criteria, the better. If the criteria are too vague or generic, every vendor will say they can satisfy your needs.

Conduct Market Research

To become informed buyers when selecting products, you should do market research to better understand the players and their offerings. A request for proposal (RFP) is a classic product evaluation tool. Although some organizations have no choice about their use, you should avoid this technique, if possible. Constructing the RFP and evaluating responses are tremendously time-consuming for the team. Meanwhile, vendors are motivated to respond to the questions in the most positive light, so the response evaluation is often more of a beauty contest. In the end, the value of the expenditure may not warrant the effort.

Evaluate a Short List of Options

Despite the plethora of products available in the market, usually only a small number of vendors can meet both functionality and technical requirements. By comparing preliminary scores from the evaluation matrix, you can focus on a narrow list of vendors and disqualify the rest. After dealing with a limited number of vendors, you can begin the detailed evaluations. Business representatives should be involved in this process if you're evaluating BI tools. As evaluators, you should drive the process rather than allow the vendors to do the driving, sharing relevant information from the architecture plan, so the sessions focus on your needs rather than on product bells and whistles. Be sure to talk with vendor references, both those formally provided and those elicited from your informal network.

If Necessary, Conduct a Prototype

After performing the detailed evaluations, sometimes a clear winner bubbles to the top, often based on the team's prior experience or relationships. In other cases, the leader emerges due to existing corporate commitments such as site licenses or legacy hardware purchases. In either situation, when a sole candidate emerges as the winner, you can bypass the prototype step (and the associated investment in both time and money). If no vendor is the apparent winner, you should conduct a prototype with no more than two products. Again, take charge of the process by developing a limited yet realistic business case study.

Select Product, Install on Trial, and Negotiate

It is time to select a product. Rather than immediately signing on the dotted line, preserve your negotiating power by making a private, not public, commitment to a single vendor. Instead of informing the vendor that you're completely sold, embark on a trial period where you have the opportunity to put the product to real use in your environment. It takes significant energy to install a product, get trained, and begin using it, so you should walk down this path only with the vendor you fully intend to buy from; a trial should not be pursued as another tire-kicking exercise. As the trial draws to a close, you have the opportunity to negotiate a purchase that's beneficial to all parties involved.

Lifecycle Data Track

In the Figure 17-1 Kimball Lifecycle diagram, the middle track following the business requirements definition focuses on data. We turn your attention in that direction throughout the next several sections.

Dimensional Modeling

Given this book's focus for the first 16 chapters, we won't spend any time discussing dimensional modeling techniques here. The next chapter provides detailed recommendations about the participants, process, and deliverables surrounding our iterative workshop approach for designing dimensional models in collaboration with business users. It's required reading for anyone involved in the modeling activity.

Physical Design

The dimensional models developed and documented via a preliminary source-to-target mapping need to be translated into a physical database. With dimensional modeling, the logical and physical designs bear a close resemblance; you don't want the database administrator to convert your lovely dimensional schema into a normalized structure during the physical design process.

Physical database implementation details vary widely by platform and project. In addition, hardware, software, and tools are evolving rapidly, so the following physical design activities and considerations merely scratch the surface.

Develop Naming and Database Standards

Table and column names are key elements of the users' experience, both for navigating the data model and viewing BI applications, so they should be meaningful to the business. You must also establish standards surrounding key declarations and the permissibility of nulls.

Develop Physical Database Model

This model should be initially built in the development server where it will be used by the ETL development team. There are several additional sets of tables that need to be designed and deployed as part of the DW/BI system, including staging tables to support the ETL system, auditing tables for ETL processing and data quality, and structures to support secure access to a subset of the data warehouse.

Develop Initial Index Plan

In addition to understanding how the relational database's query optimizer and indexes work, the database administrator also needs to be keenly aware that DW/BI requirements differ significantly from OLTP requirements. Because dimension tables have a single column primary key, you'll have a unique index on that key. If bitmapped indexes are available, you typically add single column bitmapped indexes to dimension attributes used commonly for filtering and grouping, especially those attributes that will be jointly constrained; otherwise, you should evaluate the usefulness of B-tree indexes on these attributes. Similarly, the first fact table index will typically be a B-tree or clustered index on the primary key; placing the date foreign key in the index's leading position speeds both data loads and queries because the date is frequently constrained. If the DBMS supports high-cardinality bitmapped indexes, these can be a good choice for individual foreign keys in the fact tables because they are more agnostic than clustered indexes when the user constrains dimensions in unexpected ways. The determination of other fact table indexes depends on the index options and optimization strategies within the platform. Although OLAP database engines also use indexes and have a query optimizer, unlike the relational world, the database administrator has little control in these environments.

Design Aggregations, Including OLAP Database

Contrary to popular belief, adding more hardware isn't necessarily the best weapon in the performance-tuning arsenal; leveraging aggregate tables is a far more cost-effective alternative. Whether using OLAP technology or relational aggregation tables, aggregates need to be designed in the DW/BI environment, as we'll further explore in Chapter 19: ETL Subsystems and Techniques, and Chapter 20. When performance metrics are aggregated, you either eliminate dimensions or associate the metrics with a shrunken rollup dimension that conforms to the atomic base dimension. Because you can't possibly build, store, and administer every theoretical aggregation, two primary factors need to be evaluated. First, think about the business users' access patterns derived from the requirements findings, as well as from input gained by monitoring actual usage patterns. Second, assess the data's statistical distribution to identify aggregation points that deliver bang for the buck.

Finalize Physical Storage Details

This includes the nuts-and-bolts storage structures of blocks, files, disks, partitions, and table spaces or databases. Large fact tables are typically partitioned by activity date, with data segmented by month into separate partitions while appearing to users as a single table. Partitioning by date delivers data loading, maintenance, and query performance advantages.

The aggregation, indexing and other performance tuning strategies will evolve as actual usage patterns are better understood, so be prepared for the inevitable ongoing modifications. However, you must deliver appropriately indexed and aggregated data with the initial rollout to ensure the DW/BI environment delivers reasonable query performance from the start.

ETL Design and Development

The Lifecycle's data track wraps up with the design and development of the ETL system. Chapter 19 describes the factors, presented as 34 subsystems, which must be considered during the design. Chapter 20 then provides more granular guidance about the ETL system design and development process and associated tasks. Stay tuned for more details regarding ETL.

Lifecycle BI Applications Track

The final set of parallel activities following the business requirements definition in Figure 17-1 is the BI application track where you design and develop the applications that address a portion of the users' analytic requirements. As a BI application developer once said, "Remember, this is the fun part!" You're finally using the investment in technology and data to help business users make better decisions.

Although some may feel that the data warehouse should be a completely ad hoc, self-service query environment, delivering parameter-driven BI applications will satisfy a large percentage of the business community's needs. For many business users, "ad hoc" implies the ability to change the parameters on a report to create their personalized version. There's no sense making every user start from scratch. Constructing a set of BI applications establishes a consistent analytic framework for the organization, rather than allowing each spreadsheet to tell a slightly different story. BI applications also serve to capture the analytic expertise of the organization, from monitoring performance to identifying exceptions, determining causal factors, and modeling alternative responses; this encapsulation provides a jump start for the less analytically inclined.

BI Application Specification

Following the business requirements definition, you need to review the findings and collected sample reports to identify a starter set of approximately 10 to 15 BI reports and analytic applications. You want to narrow the initial focus to the most critical capabilities to manage expectations and ensure on-time delivery. Business community input will be critical to this prioritization process. Although 15 applications may not sound like much, numerous analyses can be created from a single template merely by changing variables.

Before you start designing the initial applications, it's helpful to establish standards, such as common pull-down menus and consistent output look and feel. Using these standards, you specify each application template and capture sufficient information about the layout, input variables, calculations, and breaks, so both the application developer and business representatives share a common understanding.

During the BI application specification activity, you should also consider the applications' organization. You need to identify structured navigational paths to access the applications, reflecting the way users think about their business. Leveraging customizable information portals or dashboards are the dominant strategies for disseminating access.

BI Application Development

When you move into the development phase for the BI applications, you again need to focus on standards; naming conventions, calculations, libraries, and coding standards should be established to minimize future rework. The application development activity can begin when the database design is complete, the BI tools and metadata are installed, and a subset of historical data has been loaded. The BI application template specifications should be revisited to account for the inevitable changes to the model since the specifications were completed.

Each BI tool has product-specific tricks that can cause it to jump through hoops backward. Rather than trying to learn the techniques via trial and error, we suggest investing in appropriate tool-specific education or supplemental resources for the development team.

While the BI applications are being developed, several ancillary benefits result. BI application developers, armed with a robust access tool, will quickly find needling problems in the data haystack despite the quality assurance performed by the ETL application. This is one reason we prefer to start the BI application development activity prior to the supposed completion of the ETL system. The developers also will be the first to realistically test query response times. Now is the time to review the preliminary performance-tuning strategies.

The BI application quality assurance activities cannot be completed until the data is stabilized. You must ensure there is adequate time in the schedule beyond the final ETL cutoff to allow for an orderly wrap-up of the BI application development tasks.

Lifecycle Wrap-up Activities

The following sections provide recommendations to ensure your project comes to an orderly conclusion, while ensuring you're poised for future expansion.

Deployment

The technology, data, and BI application tracks converge at deployment. Unfortunately, this convergence does not happen naturally but requires substantial preplanning. Perhaps more important, successful deployment demands the courage and willpower to honestly assess the project's preparedness to deploy. Deployment is similar to serving a large holiday meal to friends and relatives. It can be difficult to predict exactly how long it will take to cook the meal's main entrée. Of course, if the entrée is not done, the cook is forced to slow down the side dishes to compensate for the lag before calling everyone to the table.

In the case of DW/BI deployment, the data is the main entrée. "Cooking" the data in the ETL kitchen is the most unpredictable task. Unfortunately, even if the data isn't fully cooked, you often still proceed with the DW/BI deployment because you told the warehouse guests they'd be served on a specific date and time. Because you're unwilling to slow down the pace of deployment, you march into their offices with undercooked data. No wonder users sometimes refrain from coming back for a second helping.

Although testing has undoubtedly occurred during the DW/BI development tasks, you need to perform end-to-end system testing, including data quality assurance, operations processing, performance, and usability testing. In addition to critically assessing the readiness of the DW/BI deliverables, you also need to package it with education and support for deployment. Because the user community must adopt the DW/BI system for it to be deemed successful, education is critical. The DW/BI support strategy depends on a combination of management's expectations and the realities of the deliverables. Support is often organized into a tiered structure. The first tier is website and self-service support; the second tier is provided by the power users residing in the business area; centralized support from the DW/BI team provides the final line of defense.

Maintenance and Growth

You made it through deployment, so now you're ready to kick back and relax. Not so quickly! Your job is far from complete after you deploy. You need to continue to manage the existing environment by investing resources in the following areas:

- **Support.** User support is immediately crucial following the deployment to ensure the business community gets hooked. You can't sit back in your cubicle and assume that no news from the business community is good news. If you're not hearing from them, chances are no one is using the DW/BI system. Relocate (at least temporarily) to the business community so the users have easy access to support resources. If problems with the data or BI applications are uncovered, be honest with the business to build credibility while taking immediate action to correct the problems. If the DW/BI deliverable is not of high quality, the unanticipated support demands for data reconciliation and application rework can be overwhelming.
- **Education.** You must provide a continuing education program for the DW/BI system. The curriculum should include formal refresher and advanced courses, as well as repeat introductory courses. More informal education can be offered to the developers and power users to encourage the interchange of ideas.
- **Technical support.** The DW/BI system needs to be treated as a production environment with service level agreements. Of course, technical support should proactively monitor performance and system capacity trends. You don't want to rely on the business community to tell you that performance has degraded.
- **Program support.** The DW/BI program lives on beyond the implementation of a single phase. You must closely monitor and then market your success. Communication with the varied DW/BI constituencies must continue. You must also ensure that existing implementations continue to address the needs of the business. Ongoing checkpoint reviews are a key tool to assess and identify opportunities for improvement.

If you've done your job correctly, inevitably there will be demand for growth, either for new users, new data, new BI applications, or major enhancements to existing deliverables. Unlike traditional systems development initiatives, DW/BI change should be viewed as a sign of success, not failure. As we advised earlier when discussing project scoping, the DW/BI team should not make decisions about these growth options in a vacuum; the business needs to be involved in the prioritization

process. This is a good time to leverage the prioritization grid illustrated in Figure 17-2. If you haven't done so already, an executive business sponsorship committee should be established to set DW/BI priorities that align with the organization's overall objectives. After new priorities have been identified, then you go back to the beginning of the Lifecycle and do it all again, leveraging and building on the technical, data, and BI application foundations that have already been established, while turning your attention to the new requirements.

Common Pitfalls to Avoid

Although we can provide positive recommendations about data warehousing and business intelligence, some readers better relate to a listing of common pitfalls. Here is our favorite top 10 list of common errors to avoid while building a DW/BI system. These are all quite lethal errors—one alone may be sufficient to bring down the initiative:

- **Pitfall 10:** Become overly enamored with technology and data rather than focusing on the business's requirements and goals.
- **Pitfall 9:** Fail to embrace or recruit an influential, accessible, and reasonable senior management visionary as the business sponsor of the DW/BI effort.
- **Pitfall 8:** Tackle a galactic multiyear project rather than pursuing more manageable, although still compelling, iterative development efforts.
- **Pitfall 7:** Allocate energy to construct a normalized data structure, yet run out of budget before building a viable presentation area based on dimensional models.
- **Pitfall 6:** Pay more attention to back room operational performance and ease-of-development than to front room query performance and ease of use.
- **Pitfall 5:** Make the supposedly queryable data in the presentation area overly complex. Database designers who prefer a more complex presentation should spend a year supporting business users; they'd develop a much better appreciation for the need to seek simpler solutions.
- **Pitfall 4:** Populate dimensional models on a standalone basis without regard to a data architecture that ties them together using shared, conformed dimensions.
- **Pitfall 3:** Load only summarized data into the presentation area's dimensional structures.
- **Pitfall 2:** Presume the business, its requirements and analytics, and the underlying data and the supporting technology are static.
- **Pitfall 1:** Neglect to acknowledge that DW/BI success is tied directly to business acceptance. If the users haven't accepted the DW/BI system as a foundation for improved decision making, your efforts have been exercises in futility.

Summary

This chapter provided a high-speed tour of the Kimball Lifecycle approach for DW/BI projects. We touched on the key processes and best practices. Although each project is a bit different from the next, they all require attention to the major tasks discussed to ensure a successful initiative.

The next chapter provides much more detailed coverage of the Kimball Lifecycle's collaborative workshop approach for iteratively designing dimensional models with business representatives. Chapters 19 and 20 delve into ETL system design considerations and recommended development processes.

