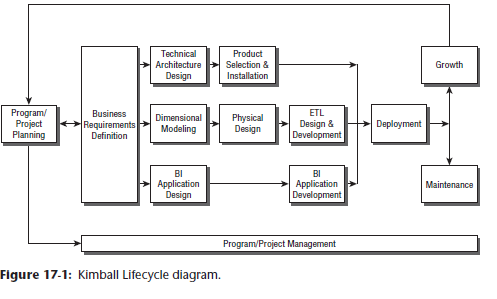
# Kimball Data Warehouse Toolkit

## Ch 17 – Kimball DW/BI Lifecycle Overview

* Now rather than focusing on Kimball dimensional modeling techniques, we turn attention to *everything else* that occurs during the course of a DW/BI design + implementation project
* Covers 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 🡪 **The Data Warehouse Lifecycle Toolkit, Second Edition** (Kimball, Ross, Thornthwaite, Mundy, Becker), Wiley, 2008
* *This chapter is a crash course drawn from the complete text, a hefty 600+ pages*
* Implementing a DW/BI system requires **tightly-integrated activities**
* Everyone on the project team (analysts, architects, designers, developers, +more) needs a high-level understanding of the complete Lifecycle
* **Concepts:**
* Kimball Lifecycle orientation
* DW/BI program/project planning + **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 first introduced, the **Kimball Lifecycle** was referred to as the ***Business Dimensional* Lifecycle**, a name that reinforced **key tenets for DW success**: **Focus on the business’s needs, present dimensionally structured data to users, and tackle manageable, iterative projects**
* These principles since become generally-accepted industry best practices
* The overall Kimball Lifecycle approach is encapsulated below



* The diagram illustrates task sequence, dependency, + 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**
* **NOTE:** Given the recent industry focus on **agile methodologies**, remember that **the Kimball Lifecycle approach + agile methodologies share some common doctrines**: **Focus on business value, collaborate with the business, + develop incrementally**
* However, **DW/BI system design + development *also* needs to be built on a solid data architecture + governance foundation, driven by the bus architecture**
* Also, **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 expected
* This module **assesses the organization’s readiness for a DW/BI initiative, establishes the preliminary scope + justification, obtains resources, + launches the program/project**
* **Ongoing project management serves as a foundation to keep the remaining activities on track**
* The 2nd major task in the roadmap focuses on **business requirements definition**.
* See a **2-way arrow between program/project planning + 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 tech won’t salvage a DW/BI environment that fails to focus on the *business***
* **Business users + their requirements have an impact on almost *every* design + implementation decision** made during the course of a DW/BI project
* In the roadmap, this is reflected by the 3 *parallel* tracks that follow:
* The top track deals with tech, as **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 + select specific products**
* Notice that product selection is NOT the 1st 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**
* Like grabbing a hammer whether you need to pound nails or tighten screws
* The middle track emanating from business requirements definition focuses on **data** + **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, + partitioning, during the physical design**
* **Last but not least, the ETL system is designed + 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 + the demanding ETL-centric activities
* Final set of tasks spawned by business requirements = **design + development of BI applications**
* **A DW/BI project isn’t done when you deliver data**
* BI applications, in the form of parameter-driven templates + analyses, will satisfy a large % of the business users’ analytic needs
* The tech, data, + BI application tracks, along with a healthy dose of education + 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**

### Lifecycle Launch Activities

#### Program/Project Planning and Management

* Not surprisingly, the DW/BI initiative begins with a series of program + 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**
* **3 factors differentiate projects** that were predominantly smooth sailing vs. those that entailed a constant struggle:
* **1) Most critical readiness factor = have a strong executive business sponsor**
* **They should have a clear vision for a DW/BI system’s potential impact on the organization**
* Optimally, business sponsors have a track record of success w/ other internal initiatives
* They should be politically astute leaders who can convince peers to support the effort
* **It’s a much riskier scenario if the CIO is the designated sponsor; prefer visible commitment from a business partner-in-crime instead**
* **2) Having a strong, compelling business motivation for tackling the DW/BI initiative**
* 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
* **3) Feasibility**
* There are **several aspects of feasibility, including technical + 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

* **Scoping** requires the **joint input of the IT organization + business management**
* The **scope** of a DW/BI project **should be both meaningful to the business organization + manageable for the IT organization**
* Should **initially tackle projects that focus on data from a *single* business process + save the more challenging, cross-process projects for a later phase**
* Avoid the **Law of Too**when scoping: **too brief of a timeline for a project w/ too many source systems + too many users in too many locations w/ 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 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**
* Need to **peel back the layers to determine the *quantifiable* impact** of improved decision making made possible by these sound bites
* **If 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 + IT communities**
* Common for the same person to fill multiple roles on the team, + assignment of named resources to roles depends on the project’s magnitude + scope, as well as the individual’s availability, capacity, + experience
* From the business side of the house, we’ll need representatives to **fill the following roles**:
* **Business sponsor 🡪** 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, **sponsor may be too far removed/inaccessible to the project team**
* In this case, **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**
* A *well-respected* person who is **highly involved in the project, communicating with the PM on a daily basi**s (Sometimes the business driver fills this role)
* **Business users 🡪** Optimally, the enthusiastic fans of the DW/BI environment
* **Need to involve them early and often**, beginning w/ the project scope + business requirements
* From there, find creative ways to **maintain their interest + involvement throughout the project**
* Remember, **business user involvement is critical to DW/BI acceptance, as 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** 🡪 responsible for determining the business needs + translating them into architectural, data, + BI application requirements
* **Data steward** 🡪 a SME**,** often the current go-to resource for ad hoc analysis
* Understands what the data means, how it is used, + where data inconsistencies are lurking
* Given the need for organizational consensus around core dimensional data, this can be a politically challenging role, as described in Chapter 4
* **BI application designer/developer** 🡪responsible for designing + 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 (PM)** 🡪a critical person who should be comfortable with + respected by business execs, as well as technical resources
* Communication and project management skills must be stellar
* **Technical architect** 🡪responsible for the overall technical architecture
* Develops the plan that ties together the required technical functionality + helps evaluate products on the basis of the overall architecture
* **Data architect/modeler** 🡪likely comes from a transactional data background with heavy emphasis on normalization
* They should embrace dimensional modeling concepts + be empathetic to the requirements of the business rather than focused strictly on saving space or reducing the ETL workload
* **Database administrator (DBA)**
* Must be willing to set aside some traditional database administration truisms, such as having only one index on a relational table.
* **Metadata coordinator** 🡪helps establish the metadata repository strategy + ensures that the appropriate metadata is collected, managed, + disseminated.
* ■ **ETL architect/designer** 🡪responsible for designing the ETL environment + processes
* ■ **ETL developer** 🡪Based on direction from the ETL architect/designer, builds + automates the processes, likely using an ETL tool.
* Again, 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 necessary Lifecycle tasks**
* A detailed task list is available at [www.kimballgroup.com](http://www.kimballgroup.com) under the Tools & Utilities tab under *The Data Warehouse Lifecycle Toolkit*, *Second Edition* book title
* Any good PM knows key team members should develop estimates of the effort required for their tasks, as the PM can’t dictate the amount of time allowed + expect conformance
* A **project plan should identify acceptance checkpoints with business reps after every major roadmap milestone + deliverable to ensure the project remains on track**
* DW/BI projects demand **broad communication**
* Although PMs typically excel at intra-team communications, they should also establish a communication strategy describing the frequency, forum, + key messaging for *other* constituencies, including the business sponsors, business community, + other IT colleagues
* Finally, DW/BI projects are vulnerable to **scope creep** largely due to a strong need to satisfy business users’ requirements
* 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)
* Say “no” (without actually saying “no” by handling the change as an enhancement request)
* **The most important thing about scope decisions = 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 w/ business users to understand their requirements + ensure their buy-in is absolutely essential to successful DW

##### Requirements Preplanning

* Before sitting down w/ business reps to collect their requirements, to ensure productive sessions:
* **Choose the Forum**
* Business **user requirements sessions are typically interwoven w/ 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, don’t ask business reps about granularity or dimensionality of their critical data
* **Need to talk to them about *what* they do, *WHY* they do it, *how* they make decisions, + *how* they hope to make decisions in the future 🡪 trying to detect issues + opportunities**
* There are **2 primary techniques for gathering requirements**: **interviews** or **facilitated sessions**, + both have advantages + disadvantages
* Interviews encourage individual participation + are also easier to schedule
* Facilitated sessions may reduce the elapsed time to gather requirements but require more time commitment from each participant
* Note: Surveys are *not* a reasonable tool for gathering requirements 🡪 flat + 2D
* 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 + the DW/BI initiative to strive for
* Can use a **hybrid approach with interviews to gather the details + then facilitation to bring the group to consensus**
* The requirements gathering forum choice depends on the team’s skills, the organization’s culture, + what the business users have already been subjected to, so one size definitely does not fit all
* **Identify and Prepare the Requirements Team**
* Regardless of the approach, you need to identify + prep the involved project team members
* If 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, it changes the meeting dynamics
* A good preference = to have a 2nd person in the room with another brain + a set of eyes + ears rather than relying on technology
* Can invite 1 or 2 additional project members (depending on the number of interviewees) as observers, so they can hear users’ input directly
* Before sitting down with business users, make sure you’re approaching the sessions with the right mindset
* Don’t presume you already know it all, as you will definitely learn more about the business during the sessions
* On the other hand, do some homework by researching available sources, such as the annual report, website, + internal organization chart.
* **Because the key to getting the right answers is asking the right questions, draf questionnaires**
* This should NOT be viewed as a *script*, but as a tool to organize your thoughts + 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 1st 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 EDW bus matrix blueprint**
* Need to understand the common data + vocabulary across core business functions to build an extensible environment.
* Within the target user community, **cover the organization *vertically***
* DW/BI project teams naturally gravitate toward the business’s power analysts.
* Although their insight is valuable, **can’t ignore senior executives + 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 reps can be the most onerous requirements task (be especially nice to the department’s admin assistants)
* Prefer to meet with executives individually
* Meeting with a homogeneous group of 2-3 people is appropriate for those lower on the organization chart
* Allow 1 hour for individual meetings + 1.5 hours for small groups
* The **scheduler** needs to allow 1/2 hour between meetings for debriefing + other necessities
* Interviewing is extremely taxing due to the focus required
* Consequently, don’t schedule more than 3 to 4 sessions in a day
* When it comes to prepping interviewees, the **business sponsor** should communicate with them, stressing their commitment to the effort + the importance of everyone’s participation
* The interviewees should be asked to bring copies of their key reports + analyses to the session
* This communication disseminates a consistent message about the project, + 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
* A 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** + 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 + *WHY* they do it**
* Simple, nonthreatening place to begin = to ask about job responsibilities + organizational fit
* A lob-ball that interviewees can easily respond to
* From there, you typically ask about their key performance metrics
* Determining **how they** **track progress + success** **translates directly into the dimensional model**, as they’re telling you about their **key business processes + facts** without you asking those questions directly.
* If you meet with a person who has more hands-on data experience, probe to better understand the business’s dimensionality
* Questions like “How do you distinguish between products (or agents, providers, or facilities, etc.)?” or “How do you naturally categorize products?” help identify key dimension attributes + hierarchies
* If the interviewee is more analytic, ask about the types of analysis currently generated
* Understanding the nature of these analyses + 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 + 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 5 reports
* The users’ questions are bound to change, + consequently, you must **resist the temptation to narrow your design focus to a supposed top 5**
* If you meet with **business executives, *don’t* dive into these tactical details**
* Instead, ask 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
* **Need to ensure the DW/BI deliverable matches the business demand + 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**
* **Interviewees must understand that just because you discussed a capability in the meeting doesn’t guarantee it’ll be included in the 1st phase of the project**
* Thank interviewees for their brilliant insights, + let them know what’s happening next + what their involvement will be.

##### Conducting Data-Centric Interviews

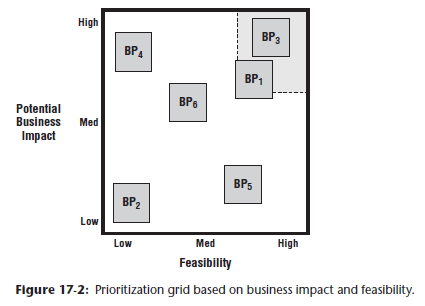
* While focused on understanding the business’s requirements, it is **helpful to intersperse sessions w/ source system data gurus or SMEs to evaluate the feasibility of supporting the business needs**
* These *data-focused* interviews are quite different from the ones just described
* **Goal = to ascertain 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 + 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**
* Must ensure everyone is on the same page about what was learned
* Also helpful if everyone reviews their notes shortly after the session to fill in gaps while the interview is still fresh
* Abbreviations + 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 DW
* 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 + project team reference materials**
* There are **2 potential levels of documentation resulting from the requirements process**
* 1) Write up each individual interview
* 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**
* 2**) 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, **begin with an executive summary, followed by a project overview covering the process used + 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, + potential benefits or impact**
* Commentary about the **feasibility of tackling each process** is also important
* As described + illustrated in Chapter 4, the processes are sometimes unveiled in an **opportunity/stakeholder matrix** to convey the impact across the organization
* In this case, **rows of the opportunity matrix identify business processes***, just like a bus matrix*
* However, in the opportunity matrix, **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 + other requirements participants**
* Inevitably you uncovered more than can be tackled in a single iteration, so you **need to prioritize**
* As discussed w/ project scope, *don’t make this decision in a vacuum*, **leverage (or foster) your partnership with the business community to establish appropriate priorities**
* The requirements wrap-up presentation is positioned as a findings review + prioritization meeting
* Participants include senior business reps (who optimally participated in the interviews), as well as the DW/BI manager + other senior IT management
* 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 below



* Grid’s vertical axis refers to the potential impact or value to the business
* 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 + feasibility**
* It’s visually obvious where to 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 🡪 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*

### Lifecycle Technology Track

* On the Kimball Lifecycle roadmap, the business requirements definition is followed immediately by **3 concurrent tracks focused on tech, data, + BI applications**, respectively, + we start with tech

#### 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 + infrastructure**
* As the EDW bus architecture supports data integration, the **architecture plan is an organizing framework to support the integration of technologies + 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) **+ 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 vs. those that will be incorporated at a later date (such as a deck and screened porch)
* ***Most important*, the architecture serves as a communication tool**
* Home construction blueprints enable the architect, general contractor, subcontractors, + 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, + outward to vendors
* Chapter 1 discussed several major components of the architecture, including ETL and BI services
* Now 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
* 1) Some teams simply don’t understand the benefits of an architecture + feel that the topic + tasks are too nebulous
* They’re so focused on *delivery* that the architecture feels like a distraction + impediment to progress, so they opt to bypass architecture design
* Instead, they piece together the technical components required for the 1st iteration with chewing gum and bailing wire, but the integration + 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
* 2) At the other extreme, some teams want to invest 2 years designing the architecture while forgetting that the primary purpose of a DW/BI environment is to solve business problems, not address any plausible (+ not so plausible) technical challenge
* **Neither end of the spectrum is healthy, + the most appropriate response lies somewhere in the middle**
* *Every* DW/BI system has a technical architecture, but the question is whether it is planned + explicit or merely implicit
* There’s a **following 8-step process to help navigate these architectural design waters**
* **1) Establish an Architecture Task Force**
* Useful to create a small task force of 2-3 people focused on architecture design
* Typically, it is the **technical architect, along with the ETL architect/designer + BI application architect/designer** who **ensure both back-room and front-room representation**
* **2) Collect Architecture-Related Requirements**
* As illustrated in the roadmap, defining the technical architecture is not the 1st 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*, listen to the business’s requirements with a slightly different filter to drive the architecture design**
* **Primary focus = uncovering the architectural implications associated w/ the business’s needs**
* Listen closely for timing, availability, + performance requirements
* Should also conduct additional interviews within the IT organization
* These are *technology-focused* sessions to understand current standards, planned technical directions, + non-negotiable boundaries
* In addition, 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
* **3) Document Architecture Requirements**
* After leveraging the business requirements process + conducting supplemental IT interviews, document your findings
* Use a simplistic tabular format, just listing each business requirement impacting the architecture, along w/ a laundry list of architectural implications
* Ex: 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
* **4) Create the Architecture Model**
* After the architecture requirements have been documented, 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, + infrastructure
* From there the team drafts + 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 can be dangerously simplistic because **significant details are embedded in the pages that follow**
* **5) Determine Architecture Implementation Phases**
* Like a dream house, likely **can’t implement ALL aspects of a technical architecture at once**
* Some are non-negotiable mandatory capabilities, whereas others are nice-to-haves
* Again, **refer back to the business requirements to establish architecture priorities** b/c you **must minimally provide the architectural elements needed to deliver the initial project**
* **6) Design and Specify the Subsystems**
* Large % of needed functionality will likely be met by major tool vendor’s standard offerings, but there’re always a few subsystems that may not be found in off-the-shelf products
* **Define these *subsystems* in enough detail, so either someone can build it for you or you can evaluate products against your needs**
* **7) 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
* A 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
* **8) 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, + 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 6 tasks associated with DW/BI product selection are quite similar to any tech selection
* **1) Understand the Corporate Purchasing Process**
* 1st step before selecting new products is to understand the internal hardware + software purchase processes
* **2) 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 w/ 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
* **3) Conduct Market Research**
* To become informed buyers when selecting products, do market research to better understand the players + their offerings
* A **request for proposal (RFP)** is a classic product evaluation tool
* Although some organizations have no choice about their use, **avoid this technique, if possible**
* Constructing an RFP + evaluating responses = **tremendously time-consuming for a 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**
* **4) 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 + technical requirements
* By **comparing preliminary scores from the evaluation matrix**, you can **focus on a narrow list of vendors** + disqualify the rest
* After dealing with a limited number of vendors, begin *detailed* evaluations
* Business reps should be involved in this process if 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 + whistles**
* Be sure to talk with vendor references, both those formally provided + those elicited from your informal network
* **5) If Necessary, Conduct a Prototype**
* After performing detailed evaluations, sometimes a clear winner bubbles to the top, often based on the team’s prior experience or relationships
* In other cases, a 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** (+ the associated investment in both time + money)
* **If *no* vendor is the apparent winner, conduct a prototype with *no more than 2 products***
* Again, **take charge of the process by developing a limited yet realistic business case study**
* **6) Select Product, Install on Trial, and Negotiate**
* Rather than immediately signing on the dotted line, preserve 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, + begin using it, so 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 Kimball Lifecycle diagram, the middle track following the business requirements definition focuses on **data**

#### Dimensional Modeling

* Next chapter provides detailed recommendations about the participants, process, + deliverables surrounding an iterative workshop approach for designing dimensional models in collaboration with business users 🡪 required reading for anyone involved in the modeling activity

#### Physical Design

* **Dimensional models developed + documented via a preliminary source-to-target mapping need to be translated into a physical database**
* **With dimensional modeling**, the **logical + physical designs bear a close resemblance**
* Don’t want the DBA to convert your lovely dimensional schema into a normalized structure during the physical design process
* Physical database implementation details vary widely by platform + project.
* In addition, hardware, software, and tools are evolving rapidly, so the following physical design activities and considerations merely scratch the surface:
* **1) Develop Naming and Database Standards**
* **Table + column names = key elements of the users’ experience**, both for navigating the data model + viewing BI applications, so they **should be meaningful to the business**
* Must also **establish standards surrounding Key declarations + the permissibility of NULLs**
* **2) Develop Physical Database Model**
* This model **should be initially built in a DEV server** where it will be used by the ETL development team
* Several additional sets of tables that need to be designed + deployed as part of the DW/BI system include **staging tables** to support the ETL system, **auditing tables** for ETL processing and data quality, + structures to support secure access to a subset of the DW
* **3) Develop Initial Index Plan**
* **In addition to understanding how the relational database’s query optimizer + indexes work, the DBA also needs to be keenly aware that DW/BI requirements differ significantly from OLTP requirements**
* **Because dimension tables have a single column PK, 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 + grouping, especially those attributes that will be jointly constrained**
* ***Otherwise*, evaluate the usefulness of B-tree indexes on these attributes**
* Similarly, **the 1st fact table index will typically be a B-tree or clustered index on the PK, + placing the date FK in the index’s leading position speeds both data loads + queries because the date is frequently constrained**
* If the **DBMS supports high-cardinality bitmapped indexes**, these **can be a good choice for individual FKs in the fact tables because they are more agnostic than clustered indexes when the user constrains dimensions in unexpected ways**
* Determination of other fact table indexes depends on the index options + optimization strategies within the platform.
* Although OLAP database engines also use indexes + have a query optimizer, unlike the relational world, the DBA has little control in these environments
* **4) 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** (Chapters 19 + 20)
* **When performance metrics are aggregated, you either eliminate dimensions *or* associate the metrics w/ a shrunken rollup dimension that conforms to the atomic base dimension**
* Because you **can’t possibly build, store, + administer every theoretical aggregation, 2 primary factors need to be evaluated:**
* 1) Think about the **business users’ access patterns derived from the requirements findings**, as well as from **input gained by monitoring actual usage patterns**
* 2) **Assess the data’s statistical distribution to identify aggregation points** that deliver bang for the buck
* **5) Finalize Physical Storage Details**
* This includes the nuts-and-bolts storage structures of **blocks, files, disks, partitions, + table spaces or databases**
* Large fact tables are typically partitioned by activity date, w/ data segmented by month into separate partitions while appearing to users as a single table
* **Partitioning by date delivers data loading, maintenance, + query performance advantages**
* The **aggregation, indexing, + 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 + aggregated data w/ 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 + development of the ETL system
* Chapter 19 🡪 factors, presented as 34 subsystems, which must be considered during the design
* Chapter 20 🡪 provides more granular guidance about the ETL system design + development process + associated tasks

### Lifecycle BI Applications Track

* Final set of parallel activities following the business requirements definition = the BI application track, where you design + develop the applications that address a portion of the users’ analytic requirements
* **You’re finally using the investment in tech + data to help business users make better decisions**
* Although some may feel that the DW should be a completely ad hoc, self-service query environment, **delivering parameter-driven BI applications will satisfy a large % 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, + modeling alternative responses, + this encapsulation provides a jump start for the less analytically inclined

#### BI Application Specification

* Following the business requirements definition, **review the findings + collected sample reports to identify a starter set of approximately 10-15 BI reports + analytic applications**
* Want to **narrow the initial focus to the *most critical* capabilities to manage expectations + 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 + consistent output look + 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 + business reps share a common understanding.
* During the BI application specification activity, **also consider the applications’ organization**
* 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, + coding standards should be established to minimize future rework**
* The application development activity can begin when the database design is complete, the BI tools + metadata are installed, + 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 + error, invest 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 it can be good to start the BI application development activity prior to the supposed completion of the ETL system
* Developers also will be the 1st 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**
* 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 2 tasks provide recommendations to ensure your project comes to an orderly conclusion, while ensuring you’re poised for future expansion

#### Deployment

* The **tech, data, + 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 + willpower to *honestly* assess the project’s preparedness to deploy**
* Deployment is similar to serving a large holiday meal to friends and relatives
* 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 DW 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, + usability testing**
* In addition to **critically assessing the readiness of the DW/BI deliverables**, you **also need to package it w/ education + support for deployment**
* **Because the user community must actually 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 + the realities of the deliverables**
* Support is often organized into a tiered structure
* 1) Website and self-service support
* 2) Provided by the power users residing in the business area
* 3) Centralized support from the DW/BI team

#### Maintenance and Growth

* You made it through deployment, but the **job is far from complete after you deploy**
* **Continue to manage an 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**
* Can’t + assume that no news from the business community is good news
* **If 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 w/ 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 + application rework can be overwhelming
* **Education**
* Must provide a continuing education program for the DW/BI system
* Curriculum should include formal refresher + advanced courses, + repeat intro courses
* More informal education can be offered to the developers + power users to encourage the interchange of ideas
* **Technical support**
* The DW/**BI system needs to be treated as a PROD environment with SLAs**
* Of course, **technical support should *proactively* monitor performance + system capacity trends**
* 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**
* Must **closely monitor + then *market* your success**
* **Communication with the varied DW/BI constituencies must continue**
* Must also **ensure that existing implementations continue to address the needs of the business**
* **Ongoing checkpoint reviews** are a key tool to **assess + 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 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 earlier
* If haven’t done so already, **an executive business sponsorship committee should be established to set DW/BI priorities that align w/ the organization’s overall objectives**
* After new priorities have been identified, go back to the beginning of the Lifecycle + do it all again, leveraging + building on the technical, data, + 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 DW + BI, some better relate to a listing of common pitfalls
* Below is a 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 tech and data rather than focusing on the business’s requirements and goals
* **Pitfall 9:** Fail to embrace or recruit an influential, accessible, + 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 query-able data in the presentation area overly complex.
* Database designers who prefer a more complex presentation should spend a year supporting business users so 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 + analytics, + 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