

Design Tip #104 Upgrading your BI Architecture

By Joy Mundy

The Kimball Lifecycle Methodology describes how to build a rich business intelligence environment to support:

- a) Publish standard reporting and scorecards: How's my business?
- b) Identify exceptions: What's unusually good or bad?
- c) Determine causal factors: Why did something go well or poorly? This step is particularly challenging because these new queries may require new data sources.
- d) Model predictive or what-if analysis: How will business look next year?
- e) Track actions: What's the impact of the decisions that were made?

Our methodology describes how to start from scratch to build this infrastructure iteratively, from project planning, requirements gathering, architecture, and design through implementation, deployment, and operations.

What do you do if you're stuck at Level (a)? What if you have an infrastructure that supports basic reporting, but is the wrong architecture to enable complex analytics or business user self-service? How do you get to where you want to go? In some ways it's easier to start from a blank slate and do it right the first time; it's easy to be a hero when you're starting from zero. But large companies, and a growing number of medium and even small companies, already have some kind of business intelligence in place. There are additional challenges in moving to a new architecture at the same time you have to maintain the existing system and users.

There are three common unsuccessful BI architectures:

- *Normalized data warehouse with no user-focused delivery layer.* The organization has invested in a BI architecture, but stopped short of the business users. The data warehouse is normalized, which means it's simpler to load and maintain, but not easy to query. Reports are written directly on the normalized structures, and often require very complex queries and stored procedures. In most cases, only a professional IT team can write reports.
- *Normalized data warehouse with mart proliferation.* A common approach to solving the problem of data model complexity is to spin off a data mart to solve a specific business problem. Usually these marts are dimensional (or at least can pass as dimensional in the dark with your glasses off). Unfortunately, they are limited in scope, contain only summary data, and are unarchitected. A new business problem requires a new mart. The better the underlying data warehouse, the easier it is to spin up a new mart. User's ad hoc access is limited to the scenarios that have been cooked into the standalone mart.
- *Mart proliferation directly from transaction systems.* The least effective architecture is to build data marts directly from OLTP systems, without an intermediate DW layer. Each mart has to develop complex ETL processes. Often, we see marts chained together as one mart feeds

the next.

In any case, the appropriate solution is to build a conformed, dimensional data delivery area. Gather business requirements and build the logical model for a Kimball conformed dimensional data warehouse.

If you already have a normalized enterprise data warehouse, analyze the gap between the business requirements and the DW contents. You might be able to build relatively simple ETL processes to populate the dimensional DW from the normalized one. For any new business processes and data, determine whether the normalized DW provides value in your environment. If so, continue to integrate and store data there, then dimensionalize and store it again in the dimensional structure.

Alternatively, you may find that it makes more sense to integrate and dimensionalize in one ETL process, and phase out the normalized DW. Once the data is in the conformed dimensional model, you'll find that business users have much greater success self-servicing and developing ad hoc queries. Some of those ad hoc queries will push up into exception, causal, and even predictive analysis, and will evolve into BI applications for the broader audience.

If you don't have a normalized data warehouse in place, you probably won't build one. This scenario is more like the "starting from scratch" approach using the Kimball Method. You'll need to gather business requirements, design the dimensional model, and develop the extract, transformation, dimensionalization, and loading logic for the enterprise dimensional data warehouse.

Arguably the biggest challenge in building an upgraded architecture is that your users' expectations are higher. You'll need to keep the existing environment in place and provide modest improvements while the new system is being developed. If you're starting from scratch, you can make users happy by rolling out the new system a little bit at a time. With a BI upgrade or replacement project, your Phase 1 scope is likely going to have to be bigger than we normally recommend to make a splash.

You need to plan for people and resources to maintain the existing environment as well as to perform the new development. We recommend that you devote a team to the new development; if the same people are trying to do the old and the new, they'll find their energies sucked into the constant operational demands of the user community. The entire group will have to expand, and the old team and new team both need business expertise and technical skills.

Once you roll out a core set of data in the upgraded environment, there are two paths you can take. You can go deeper into the initial set of data by building analytic applications that go beyond just publishing basic reports. Or, you can bring in data from additional business processes. With enough resources, you can do both at the same time.