

Design Tip #129 Are IT Procedures Beneficial to DW/BI Projects?

By Joy Mundy

Back when the world was young and data warehousing was new, projects were a lot more fun. Kimball Group consultants (before there was a Kimball Group) were usually called in by the business users, and we'd help them design and build a system largely outside the aegis of corporate IT. In the intervening years -- decades! -- data warehousing has become a mainstream component of most IT organizations. For the most part, this is a good thing: the rigor that formal IT management brings to the DW makes our systems more reliable, maintainable, and performant. No one likes the idea of a BI server sitting under a business user's desk.

However, IT infrastructure is not always helpful to the DW/BI project. Sometimes it gets in the way, or is actively obstructionist. No doubt every little sub-specialty of information technology clamors that it is somehow different or special, but in the case of DW/BI, it's really true.

Specifications. The classic waterfall methodology subscribed to by many IT organizations has you develop a painfully detailed specification that's formalized, agreed to by the business and IT, and then turned over to the developers for implementation. Changes to the specification are tightly controlled, and are the exception rather than the rule.

If you try to apply a waterfall methodology to a DW/BI project, the best you'll end up with is a reporting system. The only things you can specify in sufficient detail are standard reports, so that's what you get: a system to deliver those standard reports. Many specs include a demand to support ad hoc analysis, and sometimes include examples of specific analyses the users would like to be able to do. But within the waterfall methodology it's impossible to clearly specify the boundaries and requirements of ad hoc analyses. So the project team "meets" this requirement by plopping an ad hoc query tool in front of the database.

It's really frustrating for the business users who are asked to write or approve the specification. They know the spec doesn't capture the richness of what they want, but they don't know how to communicate their needs to IT. I was recently in a meeting with a disgruntled business user who glared at the DW manager and said "Just provide me with a system that captures all the relationships in the data." If only that one sentence were sufficient for the poor guy to design a data warehouse.

The data model takes a much more central role in the system design for a data warehouse than for a transaction system. As Bob argued in [Design Tip #123](#), the data model -- developed *collaboratively* with the business users -- becomes the core of the system specification. If the business users agree that any realistic analysis on the subject area can be met through the data in the model, and IT agrees the data model can be populated, the two sides can shake hands over the model. This is quite different from the standard waterfall approach, where the data modeler would take the spec into his cubicle and emerge several weeks later with the fully formed data model.

Naming conventions. Another place I've seen formal IT processes get in the way is in naming the entities in the data warehouse. Of course, this is much less important an issue than the specifications, but I find myself deeply annoyed by naming dogmatism. Because the DW database is designed for ad hoc use, business users are going to see the table and column names. They are displayed as report titles and column headers, so extremely long column names are problematic.

That said, naming conventions, the use of case words, and minimizing the use of abbreviations are all good ideas. But like all good ideas, temper them with reason. Taken to an extreme, you can end up with absurdly long column names like (and I'm not making this up) CurrentWorldwideTimezoneAreaIdentifier.

Although they give us something to laugh about, the real problem with absurdly long names is that the users won't tolerate them in their reports. They'll always be changing them in the report display, which means they'll use inconsistent names and introduce a new (and stupid) reason for the "multiple versions of the truth" problem. Please let your naming conventions be tempered by common sense.

Dogma. Unless it's Kimball Group dogma. I have heard all sorts of rules instituted by "someone at corporate IT." These include:

- All queries will be served from stored procedures (clearly someone doesn't understand what "ad hoc" means).
- All ETL will be done in stored procedures (All? Why?).
- All database constraints will be declared and enforced at all times (*most* of the time -- sure; but *all* the time?).
- All tables will be fully normalized (no comment).
- There will be no transformations to data in the DW (I don't have any response to this one other than a puzzled expression).

Don't get us wrong... As easy as it is to mock some practices, we believe in professionally developed and managed DW/BI systems, which usually mean IT. The advantages are huge:

- Central skills in data modeling, ETL architecture, development, and operations are greatly leveraged from one project to the next.
- Professional development, including code check-ins, code reviews, ongoing regression testing, automated ETL, and ongoing management.
- Solid bug tracking, release management techniques, and deployment procedures mean an IT-managed DW/BI system should more smoothly roll out improvements to a system already in production.
- Security and compliance.

However, if you're on a project that's suffering under an IT mandate that makes no sense to you, don't be afraid to push back.