

Design Tip #73 Relating to Agile Methodologies

By Margy Ross

I've fielded several questions recently regarding agile development methodologies. People seem to want a quick binary response: do we support and approve of agile methods or not? Unfortunately, our reaction is not so clearly black-and-white. One thing for certain is that the agile approach has enthusiastic supporters. Tackling the topic via a Design Tip might be akin to discussing someone's religion after having read a bit about it on the Internet – likely a fool-hardy proposition, but we'll jump in regardless.

First of all, what is agile software development? As with most things related to information technology, agile development takes on slightly different meanings depending on who you talk to or what you read. In general, it refers to a group of methodologies, including Extreme Programming, SCRUM, Adaptive Software Development and others, which share a common focus on iterative development and minimizing risk by delivering new functionality in short timeframes, often measured in weeks. These approaches were initially referred to as 'light-weight methodologies' in contrast to more regimented, documentation-intensive traditional methods. The term 'agile' was adopted in 2001 when a group of prominent thought leaders convened to discuss the common threads of their methodologies. The group published the Agile Manifesto (www.agilemanifesto.org) to encapsulate their shared beliefs and the non-profit Agile Alliance was established. Scott Ambler, author of several books on the subject, sums it up in a sound bite: "Agile is an iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner with 'just enough' ceremony."

There are many principals or tenets of the agile approach that resonate and tightly align with the Kimball Method's standard techniques:

- Focus on the primary objective of delivering business value. This has been our mantra for decades.
- Value collaboration between the development team and stakeholders, especially business representatives. Like the agile camp, we strongly encourage a close relationship and partnership with the business.
- Stress the importance of ongoing face-to-face communication, feedback and prioritization with the business stakeholders. While the Kimball Method encourages some written documentation, we don't want the burden to be overly onerous (or pointless).
- Adapt quickly to inevitably-evolving requirements.
- Tackle development of re-usable software in an iterative, incremental manner with concurrent, overlapping tasks. As an aside, standardizing on the 'agile' nomenclature was a marketing triumph; wouldn't you rather be agile than spiraling or stuck in a waterfall (methodology)?

So what's the bottom line? In reality, one size seldom fits all, despite what the label claims. From my vantage point, there's a time and place for agile techniques when creating a DW/BI system. They seem to naturally fit with the front-end business intelligence layer. Designing and developing the analytic reports and analyses involves unpredictable, rapidly changing requirements. The developers often have strong business acumen and curiosity, allowing them to communicate effectively with the business users. In fact, we suggest that BI-focused team members cohabitate with the business so they're readily available and responsive; this in turn encourages more business involvement. It's reasonable to deliver functionality in a matter of weeks. At the other end of the spectrum, the real-

world wrangling of the data is inherently more complex and dependent on order. While we support reducing ETL kitchen development time, the essential tasks realistically take months, not weeks, in our experience.

One final word of caution: some DW/BI development teams have naturally fallen into the trap of creating analytic or reporting solutions in a vacuum. In most of these situations, the team worked with a small set of users to extract a limited set of source data and make it available to solve their unique problems. The outcome is often a stand-alone data stovepipe that can't be leveraged by others or worse yet, delivers data that doesn't tie to the organization's other analytic information. We encourage agility, when appropriate, however building isolated data sets must be avoided. As with most things in life, moderation and balance between extremes is almost always prudent.