Business Intelligence: So where is the Intelligence?

Ask five people what Business Intelligence means to them and you will probably get ten different answers. In virtually every BI white paper it states something to this effect: “In order to ensure a 100% successful implementation for a BI project, your requirements gathering must be precise.” In other words, if you can anticipate every business question any user would have for the next five years and build your BI cubes accordingly, you will go a long way towards ensuring success.

Of course we know from real-life that this is an impossible challenge. Most businesses have no idea of what they want long-term from a BI system. Every business seems to have a BI system, but most have it for one or two specific sets of reporting requirements they would like to see met and if the BI system can satisfy these requirements, then they are happy to allocate the budget for it. Often, sadly, this is equivalent to saying, “I have a mound of sand on my lawn that is about 10 cubic feet in volume. Bring me a 20-ton Mack truck and a complete excavation crew to move it away.”

Cubes are multi-dimensional pre-indexed “replicated” (via ETL) data structures that provide rapid answers to known questions, since they were built to answer these specific questions. But if the question changes, or a new one is introduced, like “Okay, I can see that Janet is consistently the top salesperson by sales volume, but let me see who is my most PROFITABLE sales person?”

Now we need to introduce the COST_OF_SALES measure to the report and calculate who is the most profitable sales person (or most profitable product, department, region, etc.) If this information is not already in the cube (and often it is not), then the cube must now be re-designed, redeployed and completely refreshed, which could involve as many as several hundred million records. What if as a business manager, later:

- I want to analyze order size against shipping costs and then later throw in the discount applied to each order as an additional measure to do analysis against?
- How about tying returns into the equation? Surely returned products in the long run take away from profits and I get to know some data about quality control too.
- What about showing me what **ISN’T** happening? Show me the slowest moving products we have. If we have large retail stores, shelf space is at a premium. What do we stock that is not moving? If we launch a promotional campaign to get this stuff moving off the shelves, how can I track it in real time ? For example: If I have a 3 day incredible sale on all facial cosmetics in all stores in class B and C residential areas, how can I track the sales of these products in real-time?

This requires significant programming time, Administration and migration effort. These are the kinds of things that can keep BI development teams tied up for months, and in turn, keep users frustrated in waiting for what seems like an eternity. Most business executives, once they have come up with a question, are not happy to wait for months while their BI teams go redeploying the architecture in order to answer it. These executives and business analysts would be delighted if there was a quick way to modify their existing reports to show them these additional measures and have it available within the day for them to answer their questions.

Another major drawback of cubes is that most of them are refreshed with a delta data load nightly, similar to an Enterprise Data Warehouse, using an ETL process. This process can be long, and it removes any possibility of getting real-time data, since the data will be as recent as the last refresh. ETL (Extract, Transform & Load) is the process whereby

The other thing to consider with cubing technology is that it requires new skills (MDX or vendor-specific proprietary data querying interface) to be learned. Most people are not aware that MDX

is NOT the same as SQL, and the vendor-specific BI tool knowledge acquisition requires a significant investment in staffing to be able to afford those skills. Training takes a very long time, so when your skilled resource leaves for greener pastures, the company is left with a twofold problem:

1. Find a person and the budget to get them trained.
2. Find the three weeks in that person's schedule to send them off for training.

Thus most companies often opt for the even more expensive solution of hiring a top-level consultant to maintain the BI system and there is little-to-no transfer of knowledge or skills to the company's employees, increasing the company's reliance on external labor and adding to employee frustration since the in-house employees are not getting the "juicy new project" but left to maintain existing (read: old, previous century technology) systems. In addition, often this very complex BI system has a difficult and laborious process involved to answer even the simplest questions which can be equated to flying from London to Paris, via Sydney.

Another consideration is that cubes, as with data warehouses, require a new server, more storage, increased dependencies, more backup and added administration load is placed on the company. More often than not, there really is no need for these cubes. A lot of Tier 1 corporations have several hundred interdependent cubes that are refreshed daily and the organization of this data requirement is frighteningly complex (don't forget the order, dimensions are refreshed before fact tables otherwise there will be broken foreign key relationships and the cube just will not work, summary pre-aggregated data computations can only take place once all dimension and fact data is correctly loaded). What is really needed is a simple query to extract relevant data (not hundreds of gigabytes at a time and then discard most of the data) either from the OLTP system or for analyses covering longer time spans, from an EDW (Enterprise Data Warehouse) and the best way to do this is via the database. OLTP and OLAP RDBMS were made to join, it was made to process SQL queries.

With the correct tuning, this can be made a really lightweight task for the database. With some built-in caching technology and scheduling technology, where the heavy-hitting queries are run outside normal OLTP hours, this whole process can be greatly optimized without costing the company a fortune on new architecture and servers. Put a near-decent visualization component on the front end of this, and most users cannot tell what infrastructure is beneath it, just that their information is delivered very fast and in a very easy and understandable manner. This is where the intelligence comes in: being able to make sense of all the data within split seconds, being able to annotate the report, so that as it is passed up the chain of command, the person at next level above can read the comments / annotations and immediately have their attention drawn to what they should be looking at, instead of having to re-analyze the report data from scratch.

By comparison, with the SeeMoreData (SMD) approach, complex programming skills involving cube technology, Enterprise Data Warehouses, and ETL processes are often unnecessary. As new measures are introduced to a report requirement, they can be introduced and deployed within minutes and have no effect on normal processing.

Exception reporting is what really counts in most instances too. If my organization processes 100,000 transactions in a day, as an executive, do I really want to see those 100,000 transactions? (on a 2,000 page report? Do I want to upset the environmentalists for my blatant abuse of paper & ink, despite becoming the printing vendor's favorite customer?) Or do I want to see the 500 transactions that did not ship out due to insufficient inventories or the 200 transactions that were declined due to bad data or the 600 transactions that were processed against black-listed accounts or accounts in arrears. And if I could get this broken down by any demographic dimension I chose, what would this tell me about my business and how would that help me to formulate business policy?

In the case of one SeeMoreData customer, their BI manager is aware that cubes represent a level of complexity that most often can be avoided. After calling for an evaluation of SeeMoreData, all new reporting projects are now developed using SeeMoreData as their first choice. Other projects that have been ongoing for months on alternative technologies (Cognos, Brio, Hyperion) are now being considered for rewrite using SeeMoreData because of its ease of use and incredibly short development cycle.

This customer is a huge and well-known manufacturing concern, who has a huge internal training program that caters to trainee apprentices and ongoing training on new equipment in the automotive manufacturing industry. They wanted a simple way to view training enrollments, broken down by department, groups, training courses and modules, to see not only what training has been completed, but also to see what training has active enrollments, and which training any member of staff has yet to do for which they are enrolled in. Various cubing approaches were attempted, several months spent on consulting costs, with little-to-no delivery of anything the HR and Group managers wanted to see. In desperation, they looked at SeeMoreData. The SeeMoreData technicians were able to map the somewhat complex data model in under an hour back to a Microsoft Sql Server database and within 90 minutes, a successful prototype was deployed. A further 3 days of consulting saw a complete suite of interactive reports (about 50 in total) with dashboards, parameterized dynamic prompts, annotation and scheduling abilities. The longest report runs in about 15 seconds, and the customer is incredibly happy with the results.

See…More…Data

SeeMoreData

Red-E solutions
An authorized sales partner of SeeMoreData
(03) 9859-1852