

Getting on with it

Improving the data warehouse requires frequent action and the willingness to continuously learn from it. *by Dan Higgins*



Dan Higgins
Director of Teradata Warehouse
sales support

As I write this article, I'm sipping my morning coffee from a classic Teradata coffee mug. Listed on this mug are Teradata's original seven corporate values, one of which is "Getting on with it."

In the discipline of implementing data warehouses, an important aspect of "Getting on with it" means delivering a good business solution as soon as possible and not getting mired in an endless pursuit of the perfect design. It suggests we should learn to rapidly deliver good business solutions—even though they may not be perfect. Then we need to iteratively and rapidly improve those solutions based on the experience and refined understanding gained through actual use.

Time is money, whether you are measuring staff and productivity costs or financial benefits. Along the same lines, wasting time is wasting money. The sooner the business can exploit new decision support capabilities, the greater the financial reward—even while evolving those capabilities.

Consider a data warehousing project with a team of 10 people. The monthly average cost per team member is \$10,000, meaning the project run rate yields \$100,000. Let's



also assume that the monthly business benefits, such as cost savings and business process improvements, will eventually exceed \$250,000. If there is a six-month delay in providing the needed business

capability, the losses include not only the project overrun of \$600,000, but also the \$1.5 million of unrealized business benefits.

The problem is further compounded because subsequent releases build off of

earlier ones. Therefore, if the initial release is delayed six months, then it is likely that the business benefits from subsequent releases will be delayed as well.

On your mark, get set ...

I recently met with a new Teradata customer who told me his company's data warehouse was implemented and in production with hundreds of users in less time than some companies spend deciding who will be on the vendor short list. The customer's choice of Teradata technology facilitated the speed of implementation. But an equally important factor was the willingness to make a decision, commit the necessary resources and move forward, even though some mistakes were possible and future rework might be necessary.

With Teradata technology and an aggressive implementation effort, the customer was able to accelerate its original 18-month project plan and accomplish it in six months. As I understand it, the business incurred sig-

nificant savings simply from delivering early. Furthermore, the experience gained enabled the customer to continue on that accelerated pace.

Many Teradata customers understand the importance of delivering on business needs as soon as possible and evolving the data warehouse at an aggressive pace. These customers also know that successful data warehousing is a process of constant and continuous learning. To improve, you need to act. Someone once told me, "You can't steer a parked car." You need to make decisions and move on them—even if a few mistakes are made along the way. The key is to learn from those mistakes.

Now some of you may be saying, "Whoa! Are you suggesting 'Fire, ready, aim?' What about 'Haste makes waste'? Isn't it less costly to get it right the first time? What about first understanding, documenting and locking off on the detailed business requirements? Don't we need to finish the enterprise logical data model [LDM] first?"

Good questions! You are correct that good engineering practices are important. Blindly dumping source data into a data trash heap and then tearing down all the walls to give end users free rein is not recommended. Yet, on the other extreme, it is all too easy to become mired in an endless pursuit of data-model and application-design perfection to the point that we fail to deliver a good solution when the business needs it.

It's not rocket science

In the late 1980s I worked as an information systems engineer in the aerospace industry. Many of my colleagues worked on software for rockets and satellites where system requirements were very predictable since, in large part, they were governed by the laws of physics. They would spend years ensuring that the systems worked exactly as specified—you only get one shot when launching a satellite.

Yet these colleagues struggled with information systems designed for people. Unlike the laws of physics, people and business requirements are not constant. The only thing predictable about people is that their business needs will change. No matter how much time is spent discussing the proper definition of a data element or how a business question should be translated into structured query language (SQL) or the most ergonomic user interface, things will change once people use the system and learn how to exploit it.

So how do we move along at an aggressive pace while reducing the risks? Here are some suggestions to answer that question:

- > Create a shared vision and strategy. It does not need to be carved in stone; it can evolve as you learn. But without it you can quickly devolve into decision-support anarchy.
- > "Eat the elephant one bite at a time" in order to learn and gain confidence and insight for improvement.
- > Plan timely deliverables that provide new or improved business capability. Some

Progress blockers

- > **Inadequate, inappropriate or overly complex technology.** You can't deliver what the business needs if you are wrestling with technology issues.
- > **Too much focus on the technology.** Don't become so enamored with technology features that you forget your real purpose: to deliver new and better capability to the business.
- > **Design perfectionism.** Too much tuning of the data model, the application and query designs and reports can impede progress.
- > **Fear of mistakes and unforgiving end-user organizations.** If the implementers fear the consequences of a mistake, then excessive caution will kill forward momentum.
- > **Underfunding.** Do not sacrifice effectiveness in the name of cost.
- > **Over-control.** The data warehouse is for the business. Too much control will kill it.
- > **Inability to prioritize business needs.** Business users and leaders sometimes try to squeeze as much as they can into each release. But if too much is forced into the bag, it will burst, and everyone loses.
- > **Misleading measures of progress.** The job is not done when data is modeled and loaded or when the new business intelligence (BI) tool is installed. The real measure of success is when the business is using new and improved decision-support capabilities and realizing benefits. —D.H.

The sooner something is put in the hands of the end users, the sooner they will realize the benefits, and the sooner the implementers and the end users will find out what needs to be improved.

companies have a goal of delivering new capability monthly, and some even more frequently. Not all of these capabilities need to be large in size. And larger, more complex deliverables can be pipelined and developed in parallel.

- > Execute pilot projects to validate business needs, gain experience and provide tangible business benefits while minimizing wasted development costs.
- > Explore an easier approach than a pilot. Consider using direct SQL or very simple business intelligence (BI) tools to validate and sometimes even address business needs.
- > Head off a design spin cycle. When you are not converging on an agreement, either pick something and move on, or postpone that piece of the architecture or solution. Don't

let 80% of the potential business benefits get held up because you can't agree on the design for the other 20%.

- > Use post-delivery reviews that include end users to learn as much as possible and to ensure that the widest audience benefits from that education and is able to utilize it in future developments.
- > Isolate application and function-specific data structures in their own databases in order to minimize the impact of rework or redesign. Have a specific plan to migrate and integrate these applications at a later date.
- > Deliver smaller, less complex capabilities in the first two to three phases during the time you are most challenged with learning the various technologies and tools. If you

are new to data warehousing, your initial releases will be dominated by data sourcing and cleansing work.

- > Allow applications to evolve through many stages (releases). Again, requirements and priorities will likely change as the end users gain experience.
- > Learn from others. Teradata has an outstanding user community. Through The Annual PARTNERS User Group Conference & Expo, Teradata Universe conferences, Teradata discussion forums or regional Teradata User Group meetings, you can learn what has worked and not worked for others.

Now go!

As the old adage says, "Experience is the best teacher." You learn by doing.

The sooner something is put in the hands of the end users, the sooner they will realize the benefits, and the sooner the implementers and the end users will find out what needs to be improved. Value increases once the data warehouse is in place, which will enable greater insights into users' needs and a better understanding of the system's capabilities.

Furthermore, rapid delivery of capabilities may lead to a healthy improvement in the relationship between the implementers and the users. Users will have more visibility into what is being designed and built; implementers will have more frequent feedback to help them target business needs. More frequent releases will provide more opportunities for implementers and end users to celebrate progress together. So let's get on with it! **T**

Teradata: The right choice

- > **Teradata's** optimizer and strong performance provided by parallelism allow for the use of untuned queries to support pilot work.
- > **Teradata** views can provide logical tables while the underlying implementation evolves, thus reducing the impact of schema changes.
- > **Teradata** is effective with both normalized and dimensional models as it does not rely on application or function-oriented schema approaches.
- > **Teradata** can use logical databases to support and manage pilots or application-specific data structures, which are likely to change over time.
- > **Teradata's** Enterprise Data Warehouse Roadmap consists of tools and templates that can help prioritize work based on business needs.
- > **Teradata's** industry-specific logical data models (LDMs) provide a reasonable starting point for the data model, or can help get you started faster.
- > **Teradata** employs experienced data warehousing professionals who can help you move through project barriers and learning curves and assess design alternatives. —*D.H.*